SIEMENS

SIMATIC NET

SINAUT ST7 Software

System Manual

Preface

Configuration software for SINAUT ST7 1

SINAUT TD7 software package for the CPU 2

SINAUT Diagnostics and Service tool 3

SINAUT PG Routing 4

Volume 2: Software

07/2009
C79000-G8976-C222-07

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation for the specific task, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be adhered to. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

Ordernumber: C79000-G8976-C222
℗ 09/2009

Copyright © Siemens AG 2009.
Technical data subject to change

Preface

About this documentation

What's new in SINAUT ST7?

- New version "SINAUT ST7 configuration software for the PG/PC" V5.0
  - The Ethernet TIMs can be configured for communication via the MSC protocol. This allows the use of the GPRS/GSM modem SINAUT MD720-3 even in SINAUT in Internet/GPRS networks. An encrypted connection can be established from an Ethernet TIM to the Internet via a DSL modem.
  - The time slot method can now also be configured with the "SINAUT ST7 configuration software for the PG/PC" as of V5.0 for a master TIM without DCF7 receiver, if a TIM 4R-IE with an Ethernet connection to an ST7cc/ST7sc PC is used as the master TIM.

  The configuration software Version V5.0 can be used with STEP 7 as of Version V5.4 Service Pack 4.

  The configuration software version V5.0 is supported by the following operating systems:
  - Windows XP Professional SP2
  - Windows Server 2003 SP2
  - Windows Vista 32 Bit Ultimate and Business (with or without SP1)
- New firmware version V2.0 for all Ethernet TIM modules

  The new firmware supports the MSC protocol.

Replaced documentation

This manual replaces the manual release 05/2007.

Purpose of this documentation

The SINAUT ST7 system manual is split into two complementary volumes.

- **Volume 1: System & Hardware**

  This documentation will support you on your way to successful application of SINAUT ST7. This introduces you to the topic in clear and straightforward steps and provides you with an overview of the hardware components of the SINAUT ST7 station control system.

  You will be supported during the planning of network structures and topologies and will see how to install and configure individual components based on the installation guidelines. You will also find this documentation useful when installing and commissioning the SINAUT modules.

- **Volume 2: Software**

  This documentation provides you with an overview of the software components of the SINAUT ST7 station control system. You will see how individual components are configured. Diagnostics and service options are also explained.

Note

Documentation for the SINAUT ST1 system and older modules

This release of the manual "SINAUT ST7" (volume 1 + 2) no longer has detailed information on the SINAUT ST1 system and the following older modules:

- All previous TIM 3 modules: TIM 32, TIM 33, TIM 34
- TIM 4V, TIM 4VD, TIM 42, TIM 42D, TIM 43, TIM 43D, TIM 44, TIM 44D

If you require information on these modules or on SINAUT ST1, refer to release 05/2007 of this manual. You will find release 05/2007 on the Internet pages of SIMATIC NET Service & Support at the following address:

http://support.automation.siemens.com/WW/view/en/22548697

You can find the documents using the search function of SIMATIC NET Service and Support pages by entering the ID.

- ID for volume 1: 24621696
- ID for volume 2: 24619519

Validity of the documentation

This manual relates to the following software versions:

- SINAUT ST7 configuration software for the PG/PC V5.0
- SINAUT TD7 library for the CPU V2.2
- SINAUT TIM firmware V4.3.9 for the TIM 4
- SINAUT TIM firmware V2.0 for the TIM 3V-IE variants
- SINAUT TIM firmware V2.0 for the TIM 4R-IE

Further information on the Internet

You will find further information on the SINAUT products on the Internet at the following address:

http://support.automation.siemens.com/WW/view/en/22548697

There select the required information under "Entry type" (for example "Updates", "Manuals", "FAQs" etc.).

Online support

In addition to our product documentation, the comprehensive online information platform supports you in all aspects of our Service & Support at any time and from any location in the world. You will find this on the Internet at the following address:

www.siemens.com/automation/service&support

Here, you will find the following information:

- Support news, newsletter
- Product information, Product Support, Applications & Tools
- Technical Forum
- Access to other features of our Service & Support offer:
  - Technical Consulting
  - Engineering Support
  - Field Service
  - Spare parts and repairs
  - Optimization and modernization
  - Technical Support

    Expert advice on technical questions with a wide range of demand-optimized services for all our products and systems.

    www.siemens.de/automation/support-request

You will find contact addresses on the Internet at the following address:

www.automation.siemens.com/partner

SITRAIN - Siemens training for automation and industrial solutions

With over 300 different courses, SITRAIN covers the entire Siemens product and system spectrum in the field of automation and drive technology. Advanced training tailored to your needs is also available. In addition to our classic range of courses, we also offer a combination of various training media and sequences. You can, for example, use self-study programs on CD-ROM or on the Internet as preparation or to consolidate training.

You will find detailed information on our training curriculum and how to contact our customer consultants at the following Internet address:

www.siemens.com/sitrain

Table of contents

Preface **3**

1 Configuration software for SINAUT ST7 ..... **13**

1.1 Overview 13
1.1.1 The SINAUT ST7 configuration software in the SIMATIC world 13
1.1.2 Sequence of configuration of a telecontrol system 14
1.1.3 GPRS / Internet communication: Overview of configuration 15
1.1.4 Generating system data after changing the configuration of an existing system 16

1.2 Creating a project in the SIMATIC Manager 17

1.3 Creating stations and networks in network configuration 17
1.3.1 The network and station catalog 18
1.3.2 Creating networks and stations 19
1.3.3 Creating non-STEP 7 stations 20

1.4 Configuring stations in STEP 7 / HW Config 21
1.4.1 The hardware catalog 21
1.4.2 Installing racks and modules 23
1.4.3 Setting module parameters 24

1.5 Setting TIM module parameters 24
1.5.1 Overview of the tabs of the properties dialog 24
1.5.2 "General" tab 26
1.5.3 "Addresses" tab 27
1.5.4 "Special" tab 28
1.5.5 "Time Service" tab 29
1.5.6 "Interfaces" tab 37
1.5.7 "WAN Access" tab 41
1.5.8 "Options" tab 43
1.5.9 Consistency check and saving 45

1.6 Configuring networks and network nodes in STEP 7 / NetPro 45
1.6.1 Generating network attachments 46
1.6.2 Setting parameters for MPI networks 48
1.6.3 Setting parameters for Industrial Ethernet the 51
1.6.4 Setting parameters for SINAUT networks 52
1.6.5 Setting parameters for MPI network nodes 60
1.6.6 Setting parameters for Ethernet nodes 62
1.6.7 Setting parameters for WAN network nodes 64
1.6.8 Plausibility check of the network configuration 78

1.7 The SINAUT Configuration Tool 79
1.7.1 Working with the SINAUT ST7 configuration tool 79
1.7.2 Starting the SINAUT Configuration Tool 80

1.8 Configuring connections in the SINAUT Configuration Tool 81
1.8.1 Configuring SINAUT connections 81
1.8.2 Invalid Connections 84
1.8.3 Recovering lost connections 86
1.8.4 Printing connection lists 86

1.9 Subscriber administration in the SINAUT configuration tool 87

1.9.1 Subscriber list ... 88
1.9.2 Configuring the GPRS providers for MSC stations ... 90
1.9.3 The "Properties of Subscriber" dialog - parameter overview ... 92
1.9.4 "Info" tab ... 93
1.9.5 "Partners" tab ... 94
1.9.6 "Connections" tab ... 94
1.9.7 "Polling List" tab ... 98
1.9.8 "Telephone Directory" tab ... 100
1.9.9 "DB Configuration" tab ... 103
1.9.10 "SMS Configuration" tab ... 104
1.9.11 "MSC Station List" ... 110
1.9.12 "Library Info" tab ... 117
1.9.13 Printing subscriber lists ... 117

1.10 TD7onTIM software package ... 117
1.10.1 Introduction ... 117
1.10.2 Basic functions and components of TD7onTIM ... 118
1.10.3 Parameter assignment dialogs for TD7onTIM ... 120
1.10.4 Basic settings for TIM subscribers with TD7onTIM ... 122
1.10.5 Subscriber-specific parameters of TD7onTIM ... 126
1.10.6 Configuring SINAUT objects ... 129
1.10.7 Setting parameters for system objects ... 133
1.10.8 Basic parameters of the data objects ... 138
1.10.9 Channel overview and functions of channel parameter assignment ... 142
1.10.10 Mandatory parameters of the send channels ... 145
1.10.11 Mandatory parameters of the receive channels ... 148
1.10.12 Specific channel parameters of the data objects ... 149
1.10.13 Synchronization of the CPU time with TD7onTIM ... 160

1.11 Saving and generating system data ... 163
1.11.1 Saving subscriber data ... 163
1.11.2 Generating system data blocks ... 164
1.11.3 Compiling SINAUT TD7 blocks for the CPU ... 165
1.11.4 Creating SINAUT subscriber numbers as comments ... 167
1.11.5 Consistency check ... 168

1.12 Change matrix ... 169

1.13 Version information ... 171

1.14 Configuration practice ... 172
1.14.1 Downloading data blocks to the CPU ... 172
1.14.2 Downloading system data blocks to the TIM ... 172
1.14.3 Uploading stations with the Upload Station to PG function ... 173
1.14.4 Changing the MPI address of the CPU ... 173
1.14.5 Copying projects in the SIMATIC Manager ... 174
1.14.6 Avoiding time stamp conflicts ... 175

2 SINAUT TD7 software package for the CPU ... 177

2.1 Overview ... 177
2.1.1 SINAUT TD7 Library ... 181
2.1.2 Block overview ... 184
2.1.3 Changing SINAUT block numbers ... 187
2.1.4 Copying programs ... 194
2.1.5 Using online help ... 196

2.2 Principle of communication between SINAUT objects ... 199

2.3 Structure of the SINAUT user program ... 205

2.3.1 SINAUT startup program in OB100 205
2.3.2 Cyclic SINAUT program in OB1 206
2.3.3 Time-driven SINAUT program in a cyclic interrupt OB 212
2.3.4 SINAUT test routine in the programming error OB121 213

2.4 Basic blocks 214
2.4.1 FC Startup 214
2.4.2 FC BasicTask 214
2.4.3 DB BasicData 216
2.4.4 FB XCom 217
2.4.5 DB XComData 217
2.4.6 FB-PCom 218
2.4.7 DB PComData 218
2.4.8 FB BCom 218
2.4.9 DB BComData 218
2.4.10 FC Create 219
2.4.11 FC Distribute 219
2.4.12 FC Search 219
2.4.13 FC Diagnose 219

2.5 Data point typicals 220
2.5.1 Parameters you require often 223
2.5.2 ST7 binary information typical FB Bin04B_S 233
2.5.3 ST7 binary information typical FB Bin04B_R 235
2.5.4 ST7 analog value typical FB Ana04W_S 236
2.5.5 ST7 analog value typical FB Ana04W_R 245
2.5.6 ST7 counted value typicals FB Cnt01D_S and FB Cnt04D_S 246
2.5.7 ST7 counted value typicals FB Cnt01D_R and FB Cnt04D_R 249
2.5.8 ST7 command typical FB Cmd01B_S 253
2.5.9 ST7 command typical FB Cmd01B_R 255
2.5.10 ST7 setpoint typical FB Set01W_S 258
2.5.11 ST7 setpoint typical FB Set01W_R 262
2.5.12 ST7 parameter typical FB Par12D_S 266
2.5.13 ST7 parameter typical FB Par12D_R 276
2.5.14 ST7 data typical FB Dat12D_S 283
2.5.15 ST7 data typical FB Dat12D_R 287

2.6 Blocks for optional expansion 290
2.6.1 FC ListGenerator300, FC ListGenerator400 290
2.6.2 FC TimeTask 292
2.6.3 FC Trigger 294
2.6.4 FC PulseCounter 298
2.6.5 FC Safe 300
2.6.6 FC PartnerStatus 303
2.6.7 FC PartnerMonitor 304
2.6.8 FC ST7ObjectTest 310
2.6.9 FB SMS_Control 310
2.6.10 DB SMS_Data 317

2.7 Test blocks 321
2.7.1 FC TestCopy 321
2.7.2 DB TestCopyData 324

2.8 SFC / SFB system blocks used 327

3 SINAUT Diagnostics and Service tool 329

3.1 Overview of the functions and operation of the SINAUT Diagnostics and Service tool 329
3.1.1 Starting the program and types of access 329

3.1.2 Access to SINAUT subscribers and working with the diagnostics dialogs ... 332
3.1.3 Functions of the Diagnostics and Service tool ... 333

3.2 STEP 7 diagnostics ... 335
3.2.1 CPU messages ... 335
3.2.2 Module information ... 337
3.2.3 Operating mode ... 343
3.2.4 Setting the time ... 344

3.3 SINAUT diagnostics ... 345
3.3.1 TIM Diagnostics ... 345
3.3.2 TIM subscriber diagnostics ... 357
3.3.3 TIM diagnostic messages ... 362
3.3.4 TIM Message Monitor ... 366
3.3.5 TD7 CPU Diagnostics ... 368
3.3.6 TD7 Block Structure ... 369
3.3.7 TD7 Block Structure for all CPUs ... 374
3.3.8 TD7 CPU Program Comparison ... 377
3.3.9 TD7 Check of the Communication Configuration ... 379
3.3.10 TD7onTIM diagnostics ... 381
3.3.11 SDB Viewer ... 384

3.4 Service functions ... 392
3.4.1 Download SDB ... 392
3.4.2 Firmware update ... 393
3.4.3 Repair ... 395

3.5 Message protocol diagnostics ... 397
3.5.1 Testcopy DB ... 398
3.5.2 ST7cc / ST7sc protocol ... 400
3.5.3 TIM message protocol ... 400
3.5.4 Diagnostics of the TIM message protocol ... 401

3.6 Messages in the diagnostic buffer of the TIM ... 411
3.6.1 Diagnostic messages of the TIM ... 411

3.7 Messages in the diagnostic buffer of the CPU ... 423
3.7.1 SINAUT diagnostic messages of TD7onCPU ... 424

4 SINAUT PG Routing ... 429

4.1 What is PG Routing? ... 429
4.1.1 Introduction ... 429
4.1.2 Examples of configuration for PG routing ... 430
4.1.3 Range of functions of PG routing ... 433
4.1.4 Properties and restrictions of PG routing ... 434

4.2 System requirements for PG routing ... 435
4.2.1 STEP 7 ... 436
4.2.2 The SINAUT software package ... 436
4.2.3 RMOS operating system of the TIM 4 ... 436
4.2.4 TIM firmware for TIM 4 ... 437
4.2.5 Settings for SINAUT networks ... 437
4.2.6 Recompiling system blocks ... 437
4.2.7 Downloading newly compiled SDBs to TIM modules ... 438
4.2.8 Central SDB download using PG routing ... 438

4.3 Application of PG routing ... 442
4.3.1 Properties of the PG/PC interface ... 442
4.3.2 PG/PC assignment in the SINAUT network ... 443

4.3.3 PG routing in dial-up networks 446
4.3.4 Canceling the PG/PC attachment in the SINAUT network 446
Glossary 447
Index 465

1 Configuration software for SINAUT ST7

1.1 Overview

1.1.1 The SINAUT ST7 configuration software in the SIMATIC world

The SINAUT ST 7 configuration software represents the user interface for parameter assignment of SINAUT telecontrol systems. With this software, the user can implement and set the parameters for the telecontrol components in a STEP 7 project.

The following figure shows where the SINAUT ST7 configuration software fits into the overall system of the SIMATIC world. The areas with the "cloud" behind them are covered by the SINAUT ST 7 configuration software.

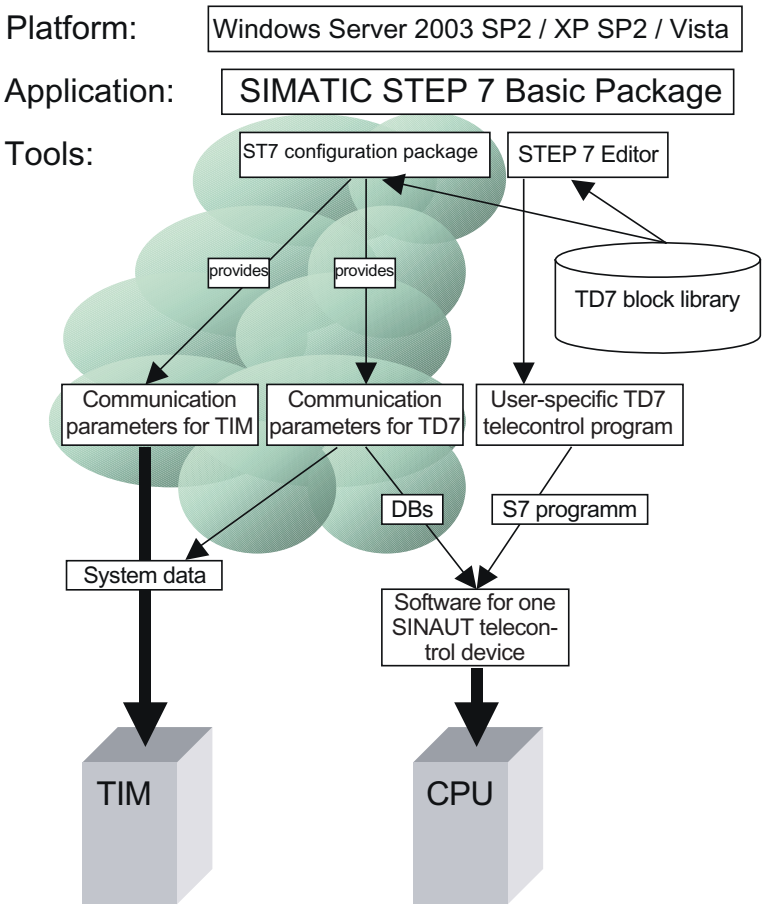


Figure 1-1 The SINAUT ST 7 configuration software within the overall system

The STEP 7 package provides the tools for configuring LANs.

The SINAUT ST7 software also allows the configuration of:

- SINAUT networks and WAN network nodes
- SINAUT TIM modules
- SINAUT connections
- Simple Internet communication

To help the user to become familiar with the SINAUT software as simply as possible, the SINAUT tools are always integrated wherever possible in the STEP 7 software. This applies in particular to the parameter assignment of the TIM modules, the SINAUT networks and WAN network nodes.

1.1.2 Sequence of configuration of a telecontrol system

The configuration of a SINAUT telecontrol system is demonstrated below step-by-step. The dialogs of the SINAUT configuration tool are also explained.

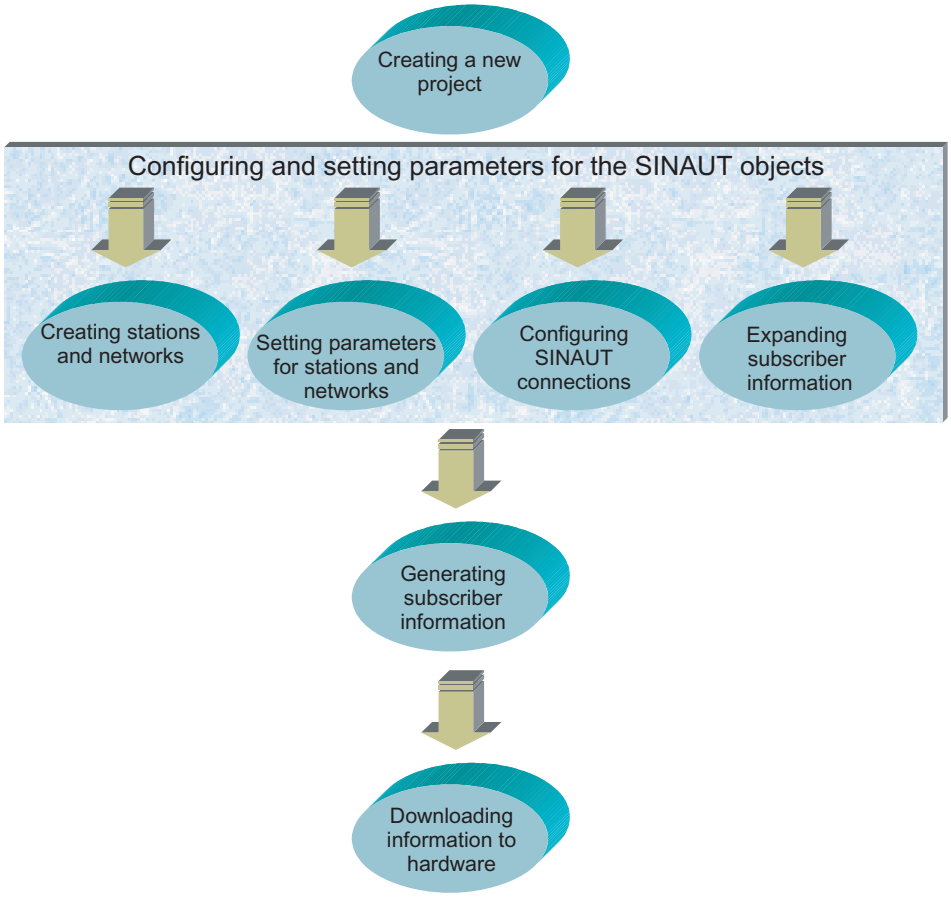

Creating a new project
Configuring and setting parameters for the SINAUT objects
Creating stations and networks
Setting parameters for stations and networks
Configuring SINAUT connections
Expanding subscriber information
Generating subscriber information
Downloading information to hardware

Figure 1-2 Sequence of configuration of a telecontrol system

When configuring a new SINAUT telecontrol system, the SINAUT configuration tool must be started after the network has been configured to allow configuration of the connections and then the data of the subscribers. Following each step, the configured data must be saved. Finally, the system data blocks for TIM and CPU modules and the SINAUT TD7 software blocks must be generated and then downloaded to the relevant modules.

Note

The language for all STEP 7 applications including the configuration tools can be changed in the SIMATIC Manager in the *Options / Customize / Language* menu.

1.1.3 GPRS / Internet communication: Overview of configuration

Alternative configurations

There are two options available for communication via GPRS / Internet:

- Highly secure communication via GPRS and Internet with the MD741-1
- Simple communication via GPRS or Internet/DSL with the MSC protocol

Communication via GPRS and Internet with the MD741-1

Communication with GPRS and a GSM network using the Internet is possible with the Ethernet TIMs and the GPRS MD741-1 modem. The MD741-1 has a VPN router with IPSec and encryption software and its own firewall.

When this is required, the Ethernet interface of the TIM is configured as a GPRS master station, GPRS node station or GPRS station, see also section "Interfaces" tab (Page 37).

Following the SINAUT configuration, the other communications parameters are configured on the MD741-1. For more information, refer to the system manual SINAUT MD741-1 in the SIMATIC NET Manual Collection that ships with the product.

Simple Internet communication with the MSC protocol

The MSC protocol is used for "simple Internet communication". The MSC protocol supports authentication of the communications partner and simple encryption of data. An MSC tunnel is established between the MSC station and MSC master station.

The following devices can be used depending on the network node type:

- MSC master station

  Only a TIM 4R-IE can be used as the MSC master station.

  Ethernet interface 1 of the TIM is connected to the Internet via a DSL modem.

- MSC station, MSC node station

  There are two alternative ways of connecting to the master station:

  – Direct connection of the MSC station/node station to the Internet

    Here, a DSL modem is connected to the Ethernet interface of the TIM.

    In this case, a TIM 3V-IE Advanced or a TIM 4R-IE can be used.

  – Internet connection of the MSC station/node station via an MD720-3 modem to a GPRS network

    Any Ethernet TIM can be used.

    During interface configuration, the WAN interface of the TIM is configured as a "dynamic Ethernet interface" (on a TIM 4R-IE, this is WAN 1).

    Only the MD720-3 can be used as the modem. The MD720-3 operates in terminal mode and as default is controlled by the application of the connected TIM. For more information, refer to the system manual SINAUT MD720-3 in the SIMATIC NET Manual Collection that ships with the product.

Configuring simple Internet communication with the MSC protocol

The networking of the interfaces is done in STEP 7 in the properties dialog of the TIM, see section "Interfaces" tab (Page 37).

You create the connection from the master station to the station in the connection configuration of the SINAUT configuration tool.

For the Internet connection of the MSC station via a GSM network, you configure the remaining MSC data in the subscriber management of the SINAUT configuration tool, see section Configuring the GPRS providers for MSC stations (Page 90) and "MSC Station List" (Page 110).

1.1.4 Generating system data after changing the configuration of an existing system

Generating system data after changing the configuration of an existing system

Changes to an existing system made in the network configuration *NetPro* or in the hardware configuration *HW Config*are saved there. The new system data blocks, on the other hand, can be generated in various configuration tools.

In the following situations, after changes have been made to the project configuration, the SINAUT configuration tool should be started to generate the system data blocks (SDB) in the *Subscriber Administration*:

- After changing the configuration of a TIM module
- After changing the configuration of a configured connection to a SINAUT subscriber
- After changing SINAUT parameters of a PC station

After changing or adding new subscribers or connections, the connection configuration and then the subscriber administration must be called in the SINAUT configuration tool to generate the SDBs there.

The activities required following a configuration change are summarized in the section Change matrix (Page 169).

1.2 Creating a project in the SIMATIC Manager

The first step in configuring a new installation is to create a new project in the STEP 7 *SIMATIC Manager*. This project serves as a directory for all the configuration data of the installation. You create the project in the SIMATIC Manager by selecting the *File / New...* menu and entering the name of the project. After creating the project, the SIMATIC Manager displays the following dialog:

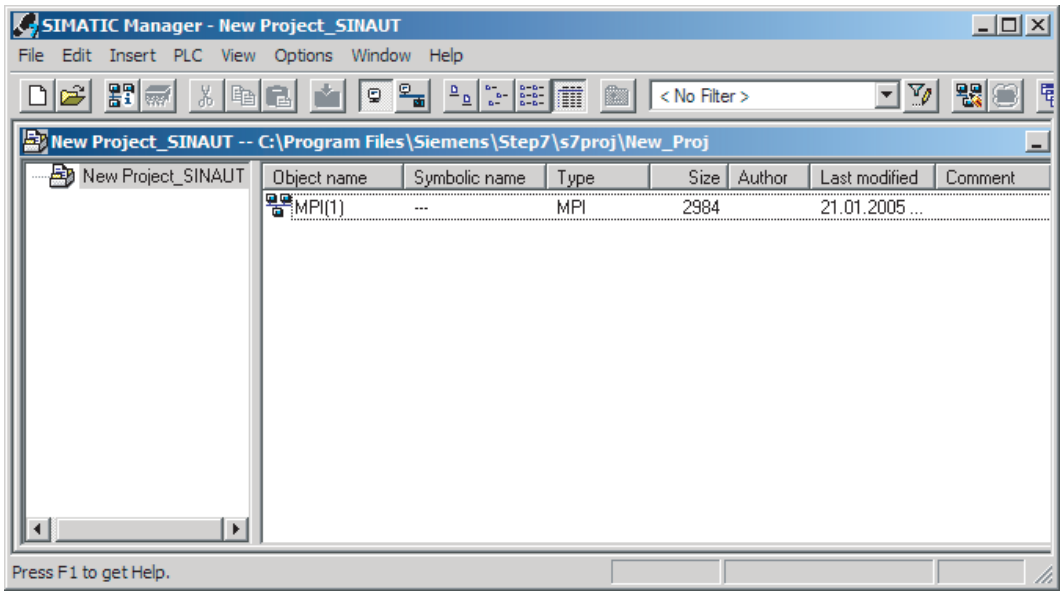


Figure 1-3 New project in the SIMATIC Manager in the *Details* view

The newly generated project is empty except for one MPI network. During configuration, the project will be gradually filled with other stations and networks. Double-clicking on the MPI network opens the *NetPro* network configuration tool.

1.3 Creating stations and networks in network configuration

The STEP 7 *NetPro* tool is used for graphic configuration of network topologies. During network configuration, networks and stations are added to a new project, given parameter settings, and interconnected. The various network types and stations are available in the network and station catalog. The basic functions and possible settings are described in the STEP 7 documentation.

1.3.1 The network and station catalog

The catalog for network configuration contains the following:

- PROFIBUS DP objects
- PROFIBUS PA objects
- PROFINET IO objects
- The possible station types
- The known network types including the SINAUT networks

When you select an object, a brief description appears in the lower part of the catalog window. The following figure shows an example of the catalog window.

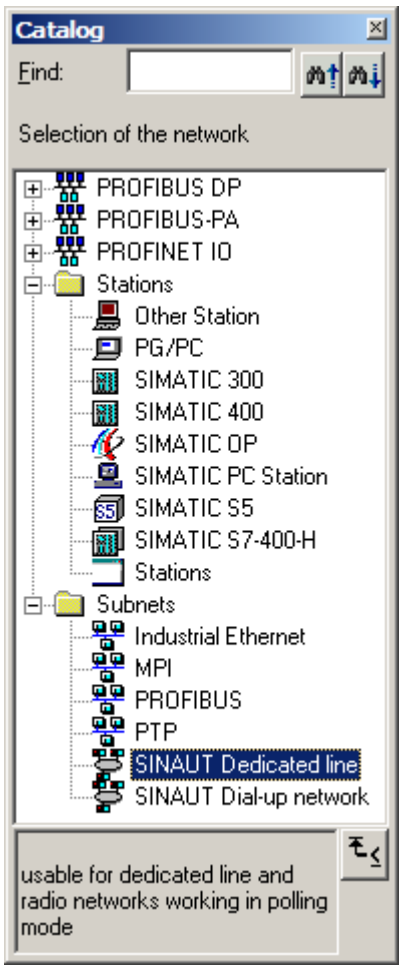


Figure 1-4 Catalog for network configuration - *NetPro*

1.3.2 Creating networks and stations

There are two ways of creating networks and stations in the network configuration:

- Double-clicking on a catalog entry creates the required object at a free position in the project area.
- Dragging a catalog entry to the project area places the object at the required position.

The position of inserted objects can be changed at any time by dragging them with the mouse.

The following figure shows a project after adding several stations and networks.

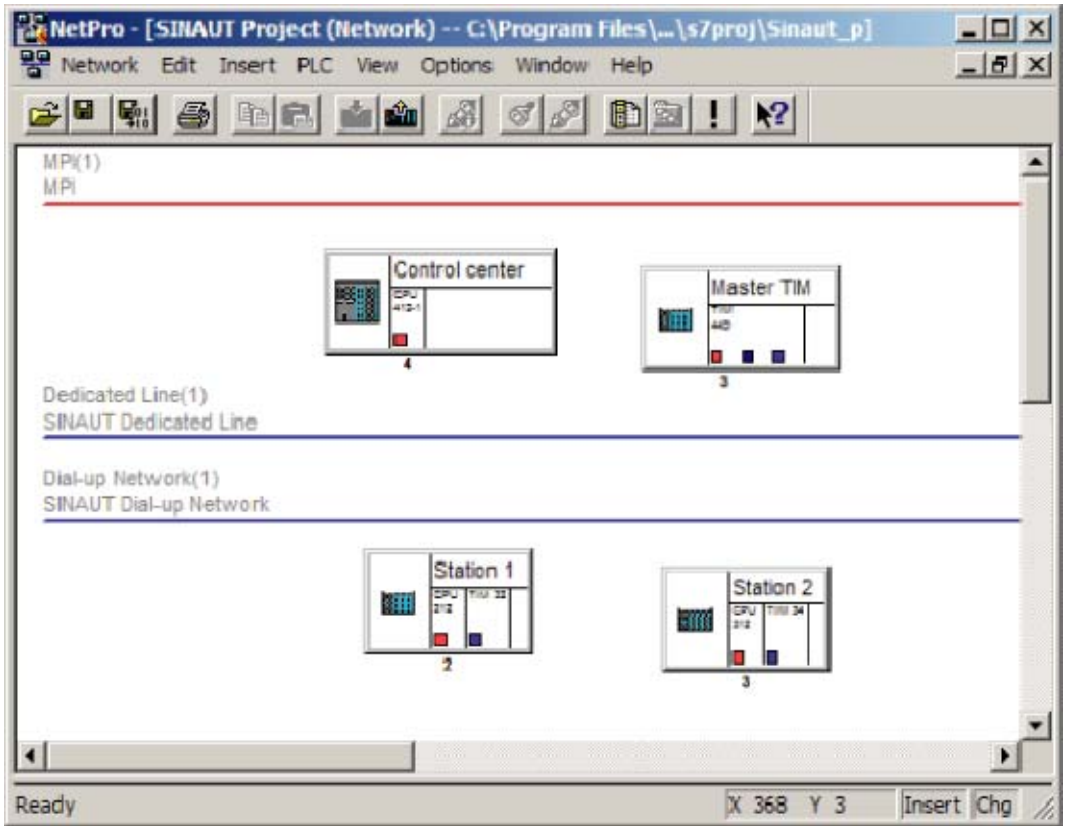


Figure 1-5 Project window for network configuration - *NetPro*

In this example, 4 stations, a dedicated line network and a dial-up network have been added to the existing MPI network. The stations are two telecontrol stations (*Station 1* and *Station 2*) and a telecontrol center consisting of the S7-400 station *Control Center* and the S7-300 as rack for accommodating *master TIMs* (stand-alone).

By selecting object with the right-hand mouse button, a context menu opens display in the available options for this object. The following functions are available for objects in the context menu:

- *Open object* (for stations only):
  This starts *HW Config* for this station.
- *Copy / Paste*:
  Copies or pastes objects.
- *Delete*:
  Deletes the selected object after confirmation in a user prompt.
- *Object properties*:
  Opens the specific properties dialog for parameter assignment of the relevant object. Here, the name and comment for an object can be entered. As long as interfaces exist, they are displayed. With networks, network wide valid parameters are set here that are relevant to the continued configuration.

1.3.3 Creating non-STEP 7 stations

Creating non-STEP 7 stations

SINAUT ST7 supports the configuration of different types of non-STEP 7 stations. These include SINAUT ST1 devices, a SINAUT ST7cc control center or an SMS center. As with the STEP 7 stations, they are created by double-clicking on the icon in the station catalog or by dragging them to the project window. The following objects must be selected for these subscribers:

- For a SINAUT ST7cc/ST7sc control center: *SIMATIC PC station*
- For an SMS center: *Other station*

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

Note

The non-STEP 7 stations listed above are used as placeholders in the network configuration. Hardware configuration of a non-STEP 7 station of the type *SMS center* is not possible since its content is unknown to STEP 7.

To attach created stations to the networks of the project, communication-compliant modules are first configured for the STEP 7 stations in *HW Config* so that the required interfaces can then be configured.

With non-STEP 7 stations, the interfaces can be created immediately.

Creating the interfaces for non-STEP 7 stations

- **SMS center:**

  An SMS center (SMSC) configured as an *other station* is configured in the *Properties* dialog available over the context menu with exactly **1** SINAUT dial-up network node.

- **ST7cc / ST7sc control center:**

  A SINAUT ST7cc or ST7sc control center configured as a *SIMATIC PC station* is equipped with the suitable communications module in hardware configuration and

attached to the master TIM over the MPI network. The station is then recognized as a SINAUT master station PC.

For detailed information on configuring a single or redundant ST7cc/ST7sc control center, refer to the SINAUT ST7cc or ST7sc documentation.

1.4 Configuring stations in STEP 7 / HW Config

The hardware configuration program *HW Config* is used to install hardware components in stations. The *HW Config* program is opened by double-clicking on one of the station icons configured in the network configuration.

The module catalog in the window on the right of HW Config contains the available objects. For SINAUT networks, these are:

- Racks
- Power supplies
- CPU modules
- SINAUT ST7 modules
- Other modules
- Applications for control centers in the SIMATIC PC station directory

The devices are installed in the station from the module catalog. Possible slot restrictions are checked and reported immediately during configuration of the object. This makes an incorrect hardware configuration impossible.

The installation rules include, for example:

- S7-300 + S7-400: Power supply permitted only in slot 1
- S7-300: CPU permitted only in slot 2
- S7-300: IM module permitted only in slot 3
- S7-300: Function modules (CPs, I/O, FMs, TIMs) permitted in slots 4 - 11
- S7-300: There must be no gaps between the modules inserted in slots 4 - 11

The installation rules for function modules are different in the expansion racks depending on the interface module (IM) with which the expansion rack is connected to the basic rack.

1.4.1 The hardware catalog

The catalog of the hardware configuration contains hardware from the following system families:

- PROFIBUS DP
- SIMATIC 300
- SIMATIC 400

- SIMATIC PC-based Control 300/400
- SIMATIC PC station

The *SIMATIC 300 / SINAUT ST7* directory contains the TIM modules of the SINAUT range.

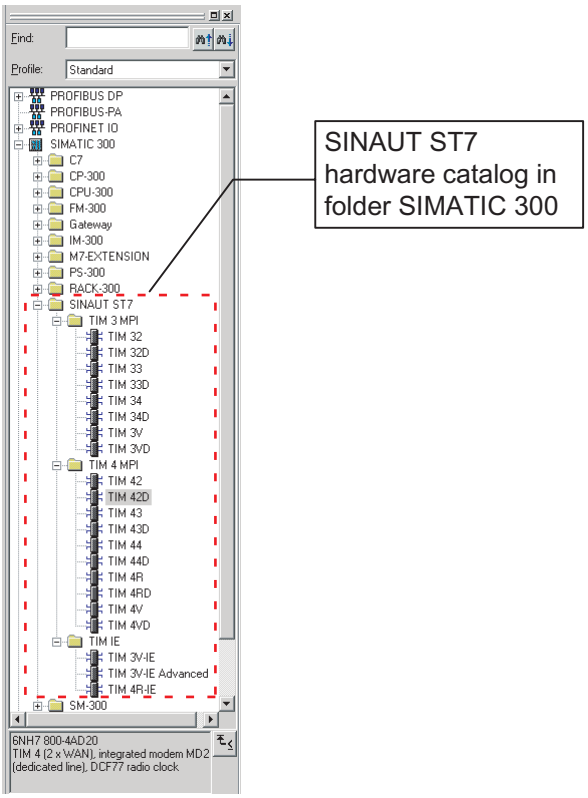


Figure 1-6 The module catalog in the hardware configuration

If you select a catalog entry, a brief explanation of the object appears below the catalog.

1.4.2 Installing racks and modules

The editing window of hardware configuration is in two parts. In the upper part of the window, you can see all the existing racks with the modules they contain. If you have a DP network, this is also visualized here. In the lower part of the window, you will see a precise listing of the modules used in the current rack along with the order number and the MPI or I/O addresses.

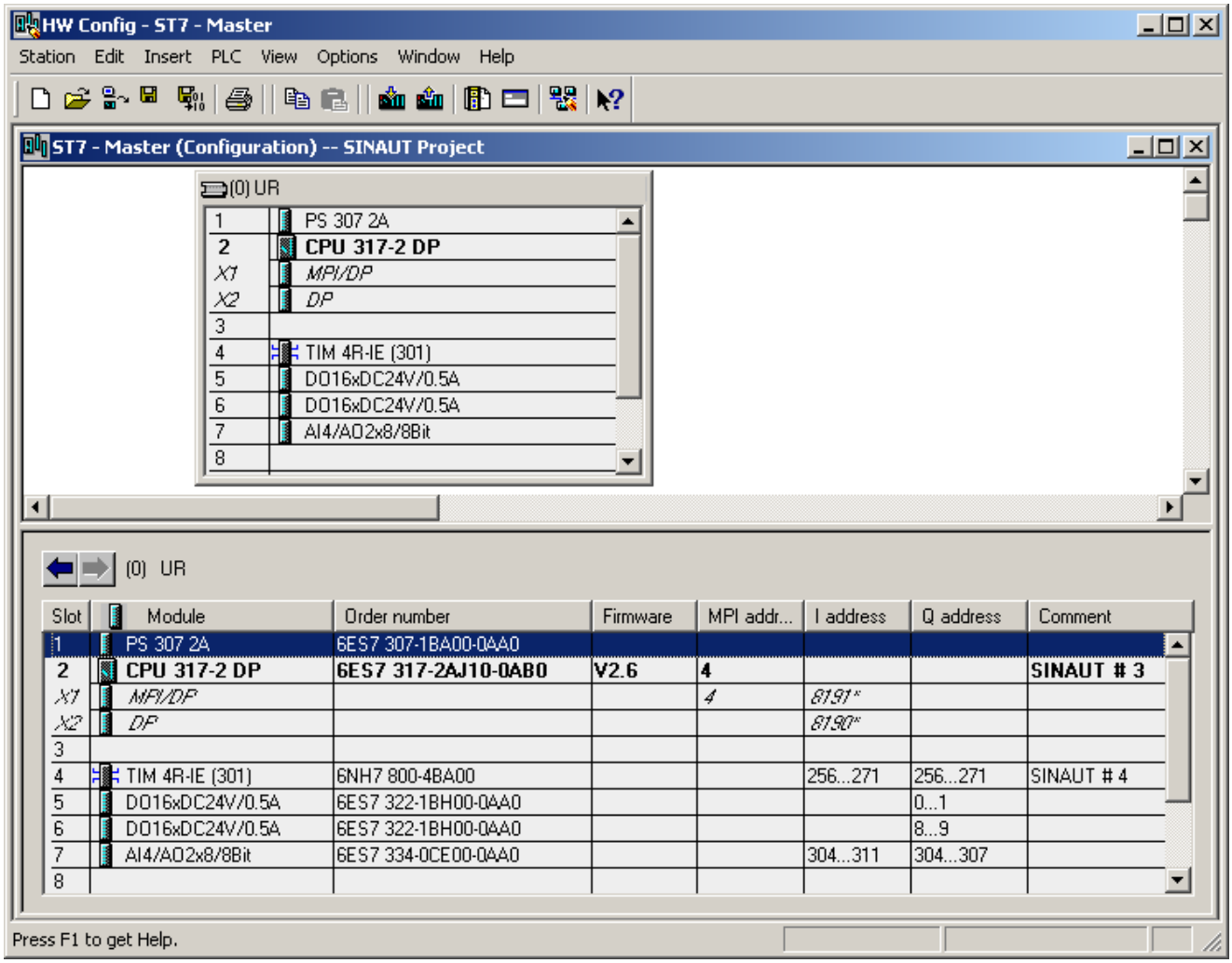

HW Config - ST7 - Master
Station Edit Insert PLC View Options Window Help
ST7 - Master (Configuration) -- SINAUT Project
(0) UR
1 PS 307 2A
2 CPU 317-2 DP
X1 MPI/DP
X2 DP
3
4 TIM 4R-IE (301)
5 DO16xDC24V/0.5A
6 DO16xDC24V/0.5A
7 AI4/AO2x8/8Bit
8
(0) UR
Slot Module Order number Firmware MPI addr... I address Q address Comment
1 PS 307 2A 6ES7 307-1BA00-0AA0
2 CPU 317-2 DP 6ES7 317-2AJ10-0AB0 V2.6 4 SINAUT # 3
X1 MPI/DP 4 8191* 8191*
X2 DP 8190*
3
4 TIM 4R-IE (301) 6NH7 800-4BA00 256...271 256...271 SINAUT # 4
5 DO16xDC24V/0.5A 6ES7 322-1BH00-0AA0 0...1
6 DO16xDC24V/0.5A 6ES7 322-1BH00-0AA0 8...9
7 AI4/AO2x8/8Bit 6ES7 334-0CE00-0AA0 304...311 304...307
8
Press F1 to get Help.

Figure 1-7 The window of HW Config with a station configuration with one SIMATIC 300 rack and various modules

Racks are either created by double-clicking on a rack entry in the catalog or by dragging the rack entry, for example a standard rail for S7-300, to the station. Since the system type SIMATIC 300 or SIMATIC 400 is already specified by the selection of the station, only suitable racks can be installed.

Modules are installed in the rack in one of two alternative ways:

- Selecting a suitable slot in the station window and double-clicking are a catalog entry or
- Dragging a catalog entry to the required slot in the station window

In both cases, the system checks immediately whether the module is permitted in the slot.

Modules can be moved to a new suitable slot at any time with the mouse.

Once a station is complete and has all the required modules, it must be saved with the *Station / Save* menu. When you close hardware configuration, a dialog opens automatically prompting you to save your entries.

When you save, the consistency of your entries is checked and a message output in the configuration errors are detected.

With the *Station / Print...* menu, you can print out the configured data of the station.

1.4.3 Setting module parameters

When you double-click on one of the modules installed in the rack in the hardware configuration, or when you select *Object Properties* in the context menu, the *Properties* dialog is opened allowing you to set parameters for a module. Here, you can adapt the properties of the particular object precisely to its requirements. The content of the *Properties* dialog depends on the module type. Only practical parameters for this type are displayed.

Since each module has a set of default parameters, it is not absolutely necessary to set parameters at this point. As an alternative, you can also open the same *Properties* dialog in NetPro.

Note

At least all the TIM modules of the project should have parameters set using the *Properties* dialog, for example to create the interfaces.

You can set parameters in the *Properties* dialog both in *hardware configuration* or in *network configuration* (NetPro).

1.5 Setting TIM module parameters

1.5.1 Overview of the tabs of the properties dialog

The parameters for a TIM module are divided among various tabs of the *Properties - TIM* dialog. The following tabs are available:

- *General*
  tab with general information and for modifying the module name or adding comments
- *Addresses*
  tab with information on I/O address areas of the CPU
- *Special*
  tab for setting parameters for an ST1 master in a dial-up network and for the diagnostic buffer

- *Time Service* tab
  for assigning parameters for time synchronization of a TIM module on the MPI bus or an Ethernet TIM (TIM 3V-IE variants, TIM 4R-IE) on Ethernet
- *Interfaces* tab
  for configuring the Ethernet and WAN interface(s)
  This tab exists only with Ethernet TIMs
- *WAN Access* tab
  for creating WAN interfaces of the TIM modules of type TIM 3/TIM 4
- *Options* tab
  with options for assigning parameters for the message memory and the message indicating a failed local subscriber to substations over a dial-up network

Note

Communication-specific parameters are entered in the *Properties* dialogs for network and network node parameter assignment. These are explained in the relevant sections.

1.5.2 "General" tab

General tab

The *General* tab informs you about the general properties of a TIM module.

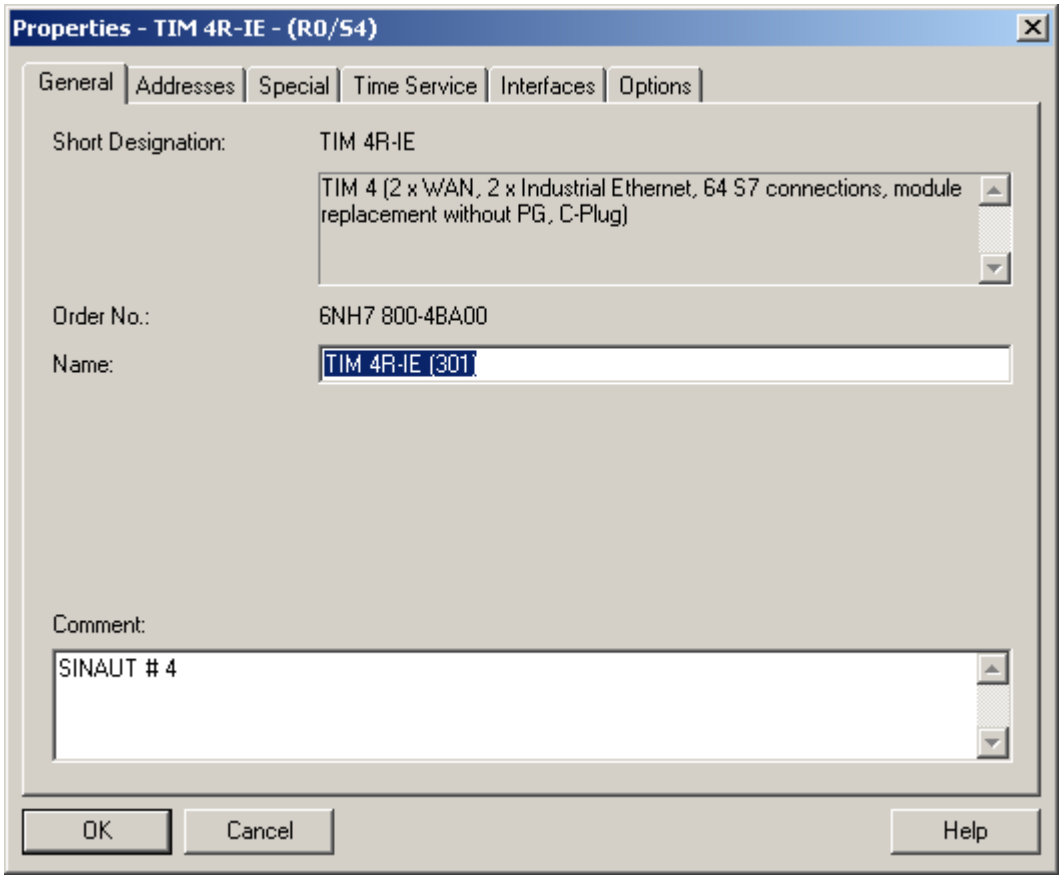


Figure 1-8 *Properties - TIM* dialog, *General* tab

This tab contains the following output boxes and parameter assignment options:

- *Short Designation:* output area This displays the module type and a brief outline of the hardware configuration.
- The *Order No.* output box displays the order number of the module.
- The *Name:* input box allows you to change the name of the module.
- As default, the *Interface* area shows the address and the networking status of the MPI interface. With the *Properties* button, you can open a dialog for setting parameters of the MPI node of the module. This is described in detail in the section on network node parameter assignment.
  Modules of the type series TIM 3 are also assigned an MPI address as default, even if this does not physically exist. If you click the *Properties* button, the parameters of the MPI interface are not available for these modules in the next dialog.

- The *Comment:* input box allows you to enter comments, for example on the purpose of the module.
  The SINAUT subscriber number of the TIM module that can be generated as a comment in the *Subscriber Administration* of the SINAUT configuration software is then displayed in this comment box.

1.5.3 "Addresses" tab

Addresses tab

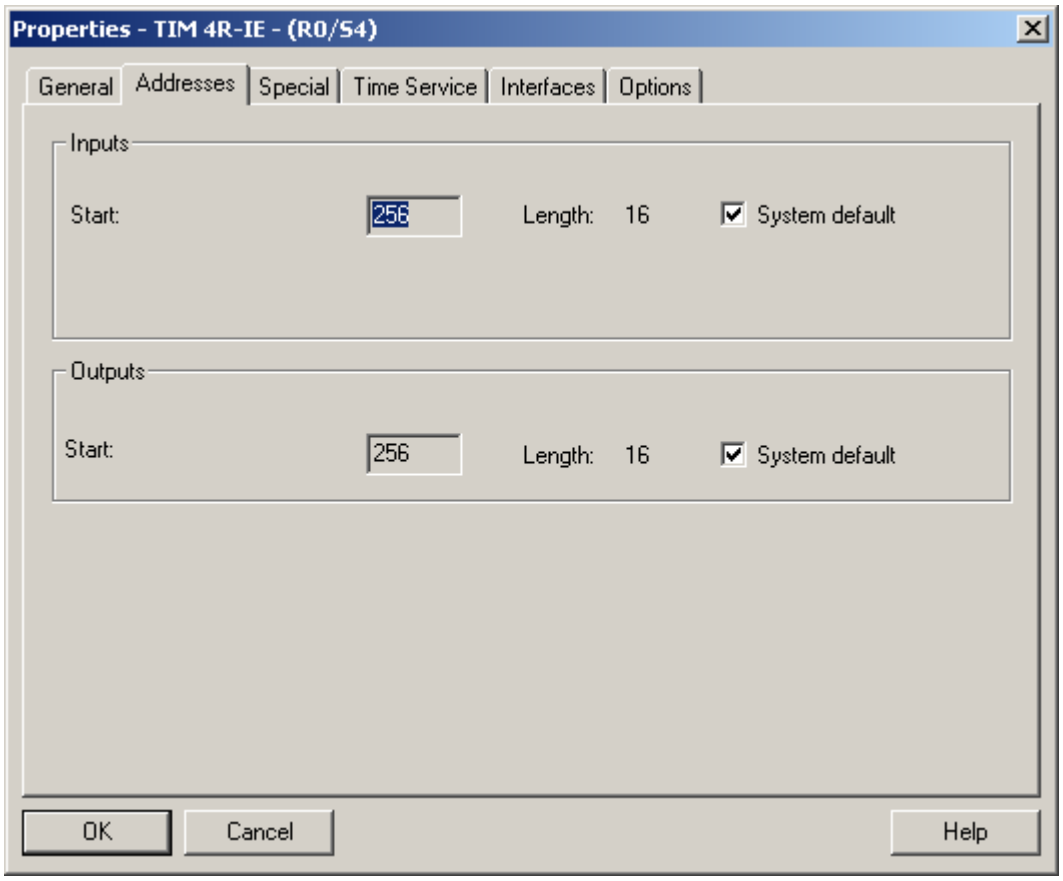


Figure 1-9 *Properties - TIM* dialog, *Addresses* tab

The Addresses tab provides information on the address areas occupied by the TIM module in the I/O from the perspective of the CPU.

These addresses are only relevant to you when the SINAUT program is configured on the TIM (TD7onTIM, possible with Ethernet TIMs) and when the CPU is supplied with the date and time by the TIM. In this case, the TIM supplies the time data to the inputs specified here. This is described in detail elsewhere (refer to the section: Synchronization of the CPU time with TD7onTIM)

The start address and length of the address ranges are assigned by the system. As an alternative, you can change the inputs and outputs by disabling the system selection option

and entering the start address in the input box manually. Since the addresses are always set consistently by the system and are not generally used, it is not normally necessary to make a change.

1.5.4 "Special" tab

Special tab

The *Special* tab shows the SINAUT subscriber number of the TIM module and you can set the size and configuration of the diagnostic buffer.

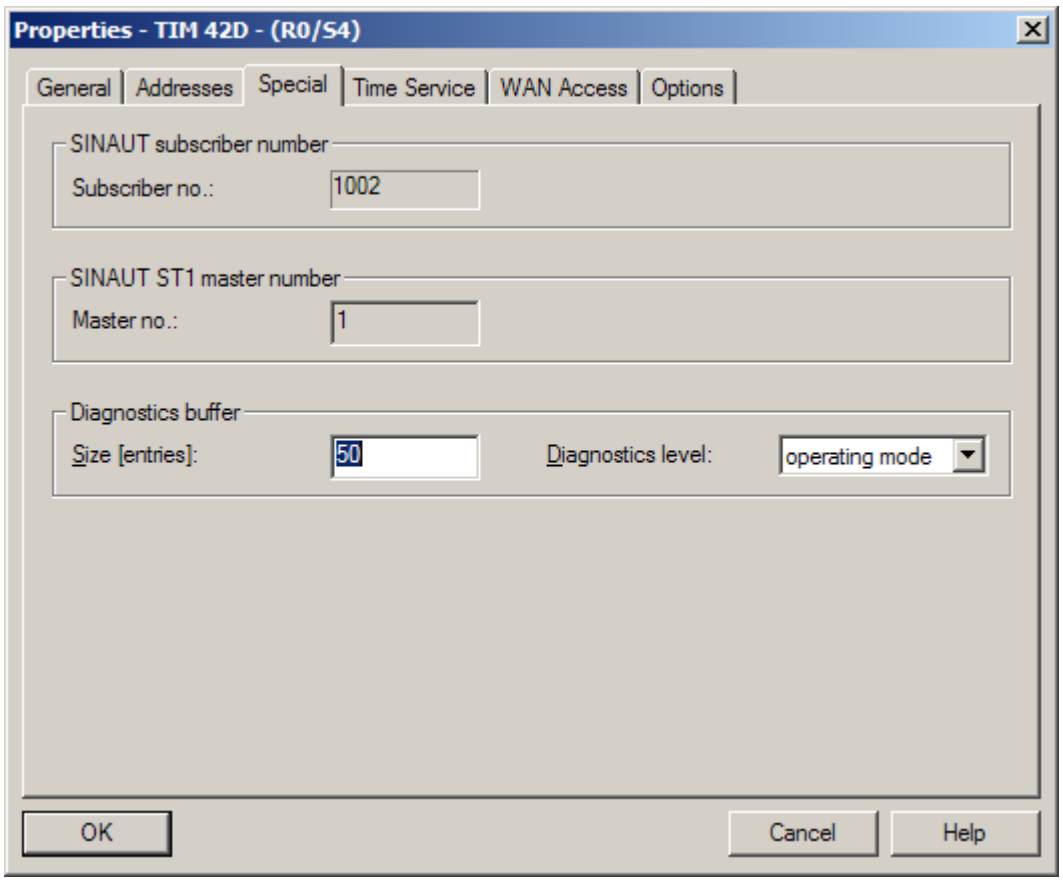


Figure 1-10 *Properties - TIM* dialog, *Special* tab

This tab contains the following parameter assignment options:

- *SINAUT subscriber number* box:
  The project-wide unique SINAUT subscriber number is displayed here. For more data of information, refer to the configuration of the *subscriber administration* in the SINAUT configuration tool.
- *SINAUT ST1 master number* box:
  In ST1 dial-up networks, the SINAUT ST1 master number must be specified for the TIM modules. For more information on SINAUT ST1, refer to release 05/2007 of the manual.

- *Diagnostics buffer size* box:
  The diagnostics buffer is organized as a circulating buffer and can hold the specified number of messages.
  Range of values: 10 ... 100
  Default value: 50
- *Diagnostics level*:
  The diagnostics messages required for normal operation are generated in the *Operating mode*. In *Service mode*, additional diagnostics messages are generated.

1.5.5 "Time Service" tab

Time Service tab

Note

Here, it is not possible to set the time synchronization of the network attachments to a SINAUT dedicated line or a SINAUT dial-up network (RS-232/RS-485 port on the TIM). You make these settings in the properties dialog of the relevant dedicated line or dial-up network (refer to the section: Setting parameters for SINAUT networks, Time Services tab). There is no setting per network node in this tab because the hierarchical distribution of the time is specified automatically during parameter assignment (master/node station/station).

In the *Time Service* tab of a TIM module, you decide how the TIM will react to time synchronization on its interfaces:

- Time synchronization on the Ethernet interface(s)
- Time synchronization on the S7-300 backplane bus when the TIM is inserted in an S7-300
- Time synchronization on the MPI bus (with the TIM 4)

The figure below shows the tab for an Ethernet TIM (TIM 3V-IE variants and TIM 4R-IE).

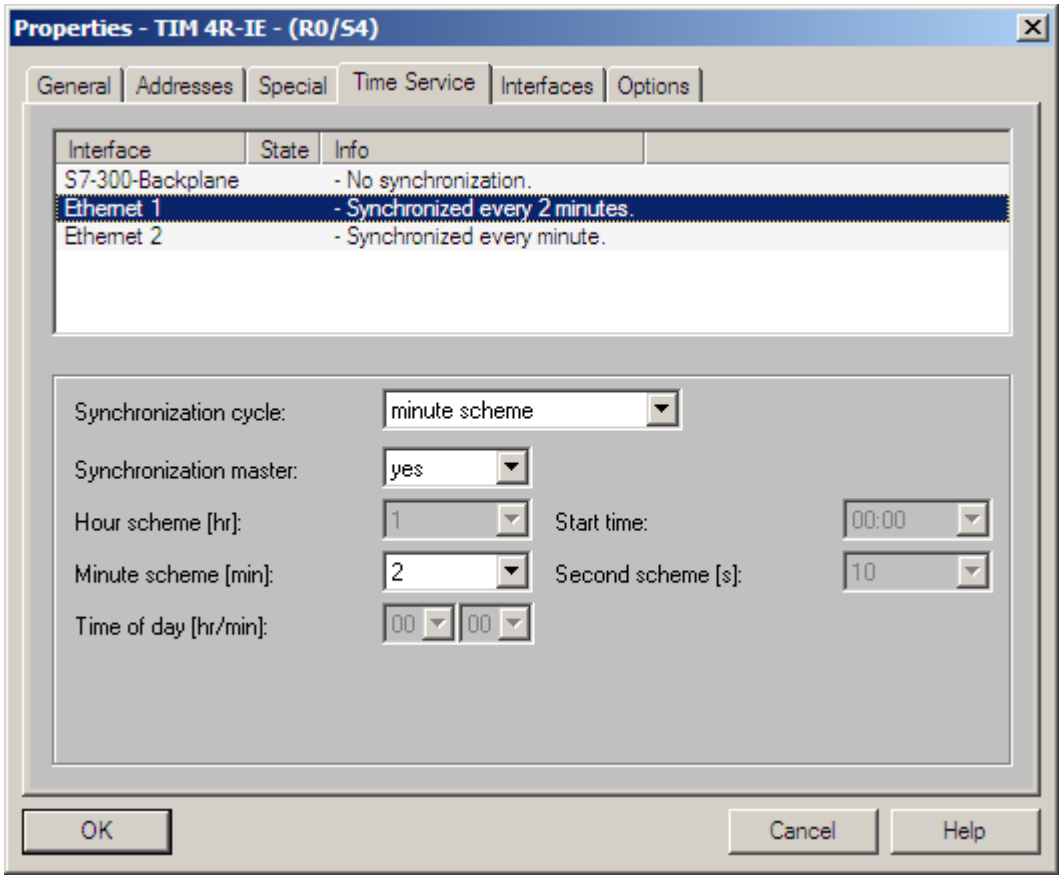


Figure 1-11 *Properties - TIM* dialog, *Time Service* tab

The interfaces for which you can set parameters are listed in an overview box. If you click on one of the interfaces, the parameters for time synchronization appear below the box for the interface.

Note

In the overview box of the TIM 4R-IE, the "S7-300 backplane bus" interface is always displayed. Parameters can, however, only be set for this interface if the TIM 4R-IE is inserted in an S7-300 as a CP.

The overview box is not available for modules without an Ethernet interface. Here, you can only set parameters for time synchronization on the MPI bus or S7-300 backplane bus. The same parameters are displayed as for an Ethernet TIM if the "S7-300 backplane bus" is selected there.

If a TIM is inserted in an S7-300 as a CP, time synchronization on the "S7-300 backplane bus" specifies when the time synchronization of the local S7-300 CPU is performed. If other TIMs are inserted in the same S7 rack, these are also synchronized at the same intervals as set here.

For a TIM 4 with MPI interface, the parameter settings for synchronization of the SINAUT nodes attached to the MPI bus apply (PCs, S7-300 and S7-400 CPUs and any other TIM 4 modules connected to the MPI bus).

The following rule applies if there are several TIMs in the S7-300 rack or several TIMs on the MPI bus:

1. Time synchronization must be enabled for all TIMs and set to the same time interval.

   After startup, only one of the TIMs will actually behave as the time master. This is negotiated automatically by the TIMs. The TIM acting as time master synchronizes all the local SINAUT nodes known to it. The other TIMs act as slaves and allow themselves to be synchronized by the current master. If the TIM acting as master fails, one of the other TIMs automatically takes over the time master function until the failed master TIM is available again.

While the "time master" or "time slave" roles are negotiated automatically on the S7-300 backplane bus or MPI, with the Ethernet interfaces of the TIM, the role of master or slave must be specified explicitly. The following rules apply:

1. If the Ethernet port of the TIM is connected to an Ethernet on which there is also an ST7cc or ST7sc PC, the PC in this network is always time master, in other words, the relevant Ethernet port of the TIM must be set to the "slave" function.
2. In an Ethernet network without ST7cc or ST7sc PC, the Ethernet port of one of the TIMs must be set to master and all others to slave. If the master function is set for more than one TIM on the Ethernet network, an error message is generated during the verification performed in the SINAUT node management.
3. Each Ethernet port to be synchronized by a master must be enabled as a slave. Otherwise synchronization is not accepted on this port. The setting of the synchronization interval or time of a slave should be identical to that of the master on the Ethernet network because the slave monitors whether or not the synchronization takes place at the specified intervals or at the specified time. Setting a shorter interval or a different time would lead to error messages in the diagnostic buffer of the TIM.

Note

The error message is not exactly coordinated with the interval or the point in time.

- With an interval, the error message comes after 2.5 times the set interval. Example: At an interval of 2 hours, the error message is entered only after 5 hours.
- If a specific time is selected, a tolerance of 2.5 hours is allowed before the error is signaled.

In the following example, meant to illustrate the two sections of a SINAUT project, shows where and which time synchronization setting must be made.

Example of time synchronization

It is assumed that all SINAUT nodes in the network need to be synchronized.

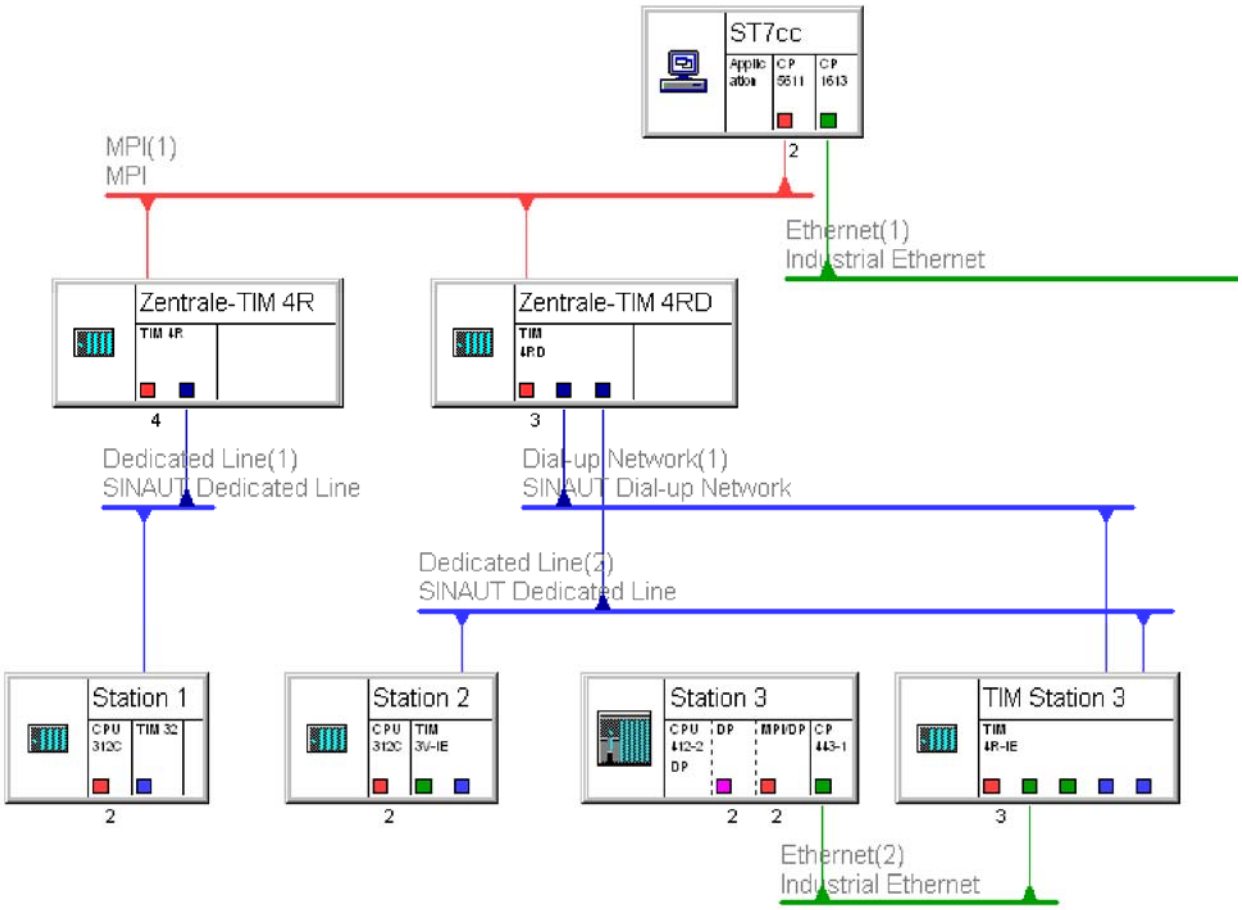


Figure 1-12 Example of time synchronization: Network section 1 - MPI / classic WAN

- **ST7cc**
  No settings are necessary here.
- **Master TIM 4R / master TIM 4RD**
  Time synchronization on the MPI bus must be enabled for both TIMs. An interval of 1 minute is recommended.

Note

ST7cc and ST7sc expect time synchronization on the MPI bus at intervals of 1 minute or less. Longer intervals cause error messages.

In this example, the "master TIM 4RD" takes over the master function after startup because it has a DCF77 receiver. Once this TIM has received a valid time of day, the ST7cc PC and the "master TIM 4R" are synchronized over MPI. If the TIM with a DCF77 receiver fails, the "master TIM 4R" can take over the master function.

Both TIMs synchronize the stations connected over a dedicated line or dial-up network by synchronizing the TIMs in the stations that, in turn, supply their own CPU. In these networks, you do not need to make settings for the TIMs. These TIMs obtain their parameters from the time parameter settings made centrally for the particular SINAUT network (dedicated line or dial-up network) refer to the section: Setting parameters for SINAUT networks, Time Services tab).

- **Station1, TIM 32**
  Time synchronization on the S7-300 backplane bus must be enabled for this TIM; in other words, the TIM then supplies the S7-300 CPU with the current time over the backplane bus. An interval of 1 minute is recommended.

Note

If a TIM synchronizes an S7-300 CPU over the backplane bus, no synchronization settings are necessary for the S7-300 CPU in HW Config. The SINAUT software on the CPU (TD7onCPU, FC TimeTask) handles the synchronization by using the synchronization message of the TIM to set the CPU clock.

- **Station 2, TIM 3V-IE**
  Here, there are two situations to be taken into account:
  - A SINAUT program (TD7onCPU) is running on the CPU:
    You set the time synchronization on the S7-300 backplane for the TIM.
  - The SINAUT program is configured on the TIM 3V-IE (TD7onTIM):
    You do not need to enable time synchronization on the TIM.
    Although there is no SINAUT program on the CPU, the CPU can nevertheless by supplied with the time of day when necessary. In this case, the TIM supplies the time data to its inputs. This is explained in detail elsewhere (refer to the section: Synchronization of the CPU time with TD7onTIM).
- **TIM Station 3**
  This TIM handles the SINAUT communication for the S7-400 "Station 3". Here, you will need to enable time synchronization on the relevant Ethernet interface of the TIM as master. An interval of 1 minute is recommended.
- **Station 3**
  No settings need to be made for the S7-400 CPU nor for the CP 443 in HW Config. The SINAUT software on the CPU (TD7onCPU, FC TimeTask) handles the synchronization by using the synchronization message of the TIM to set the CPU clock.

The next figure shows a further excerpt from the sample project which will be used to explain further details on synchronization, particularly in an Ethernet network.

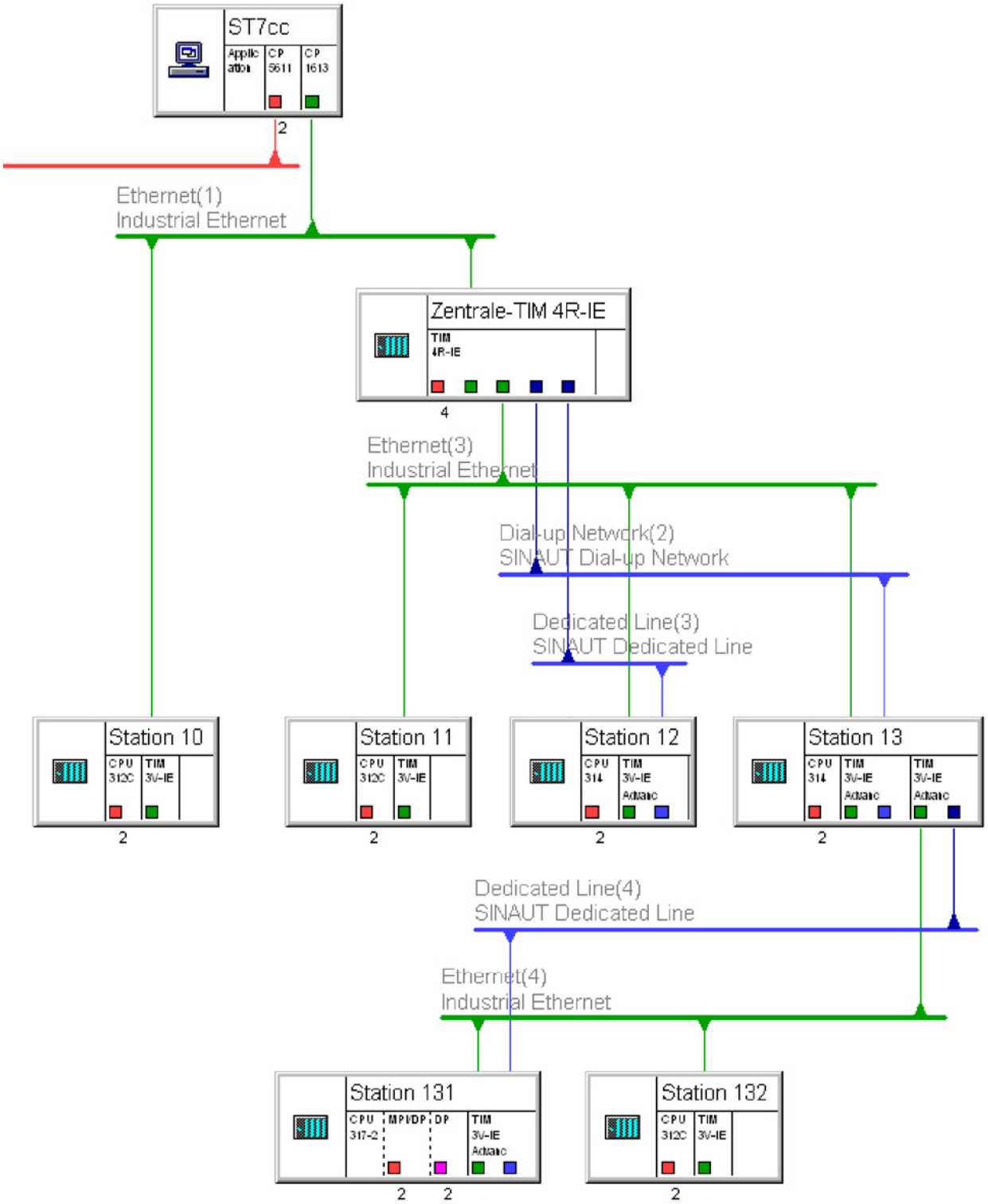


Figure 1-13 Example of time synchronization: Network section 2 - Ethernet

- **ST7cc**
  No settings need to be made in an Ethernet network in which ST7cc (or ST7sc) is always time master.
- **Station 10, TIM 3V-IE**
  This TIM is connected directly to ST7cc, the time master, over Ethernet. You will therefore need to enable the Ethernet port of the TIM as slave for time synchronization. An interval of 1 minute is recommended.

  For the CPU in station 10, you may want to enable synchronization on the S7-300 backplane bus. Refer to the notes above in the section "Station 2, TIM 3V-IE".

  Note

  If the station is connected to an Ethernet network for which fees are charged, for example via GPRS, it may be more economic to set an interval longer than 1 minute.

- **Master TIM 4R-IE**
  This TIM has two networked Ethernet accesses. Make the following settings.
  - On Ethernet(1):
    There is an ST7cc computer (= time master) on this Ethernet network. Enabling the interface as time slave. An interval of 1 minute is recommended.
  - On Ethernet(3):
    Enable the interface of the TIM as time master on this Ethernet network. Here, you can set an interval different from the interval for the slave interface on Ethernet(1).

  Apart from synchronizing the stations connected to Ethernet(3), the TIM also supplies the stations in the dedicated line or dial-up network by synchronizing the TIMs in these stations that, in turn, supply their CPUs. In these networks, you do not need to make settings for the TIMs. These TIMs obtain their parameters from the time parameter settings made centrally for the particular SINAUT network (dedicated line or dial-up network) refer to the section: Setting parameters for SINAUT networks, Time Services tab).
- **Station 11, TIM 3V-IE / station 12, TIM 3V-IE Advanced**
  The TIMs in both stations are attached to Ethernet(3) in which the "master TIM 4R-IE" is enabled as time master. This means that you will need to enable both TIMs as time slaves. The interval should be identical to that on the time master on Ethernet(3).

  For the CPU in station 11 or 12, you may want to enable synchronization on the S7-300 backplane bus. Refer to the notes above in the section "Station 2, TIM 3V-IE".
- **Station 13, two TIM 3V-IE Advanced modules**
  This station functions as a node station. Each of the two TIMs has a network access to Ethernet that you enable as time slave for the TIM on Ethernet(3) and as time master on the other TIM on Ethernet(4).

  To allow the TIM connected to Ethernet(4) to adopt the role of time master, it must be synchronized by the TIM connected to Ethernet(3). The TIMs are synchronized over the S7-300 backplane bus. To allow this, you will need to enable time synchronization over the S7-300 backplane bus on both TIMs.

  The hierarchical structure of the time synchronization network is continued here. If necessary, you can set an interval for the master interface (right-hand TIM) that differs from the interval for the slave interface (left-hand TIM).

Parameter settings for time synchronization

You can set the following options for the *Synchronization cycle* parameter:

- No synchronization:
  There is no time synchronization on the relevant network.
- Hour scheme:
  The number of hours between synchronization activities can be set in the "Hour scheme" drop-down list box.
  - Start time:
    If the cycle for time synchronization is longer than 1 hour, you can set a start time for time synchronization in the "Start time" drop-down list box.
- Minute scheme:
  The number of minutes between synchronization activities can be set in the "Minute scheme" drop-down list box.
- Second scheme:
  The number of seconds between synchronization activities can be set in the "Second scheme" drop-down list box.
- Time of day:
  Synchronization takes place once a day. Set the time of day for the synchronization in the "Time of day" drop-down list box (for example 01:00).
- Synchronization master (only for Ethernet port)
  Here, you can decide whether the TIM module adopts the master role for time synchronization (setting "yes") or not (setting "no"). If "no" is set, the TIM is a time slave.

  You will find more information on setting the time master or slave in the explanations above.

1.5.6 "Interfaces" tab

Interfaces tab

The *Interfaces* tab is available only for Ethernet TIMs. It displays a list of interfaces of the TIM module.

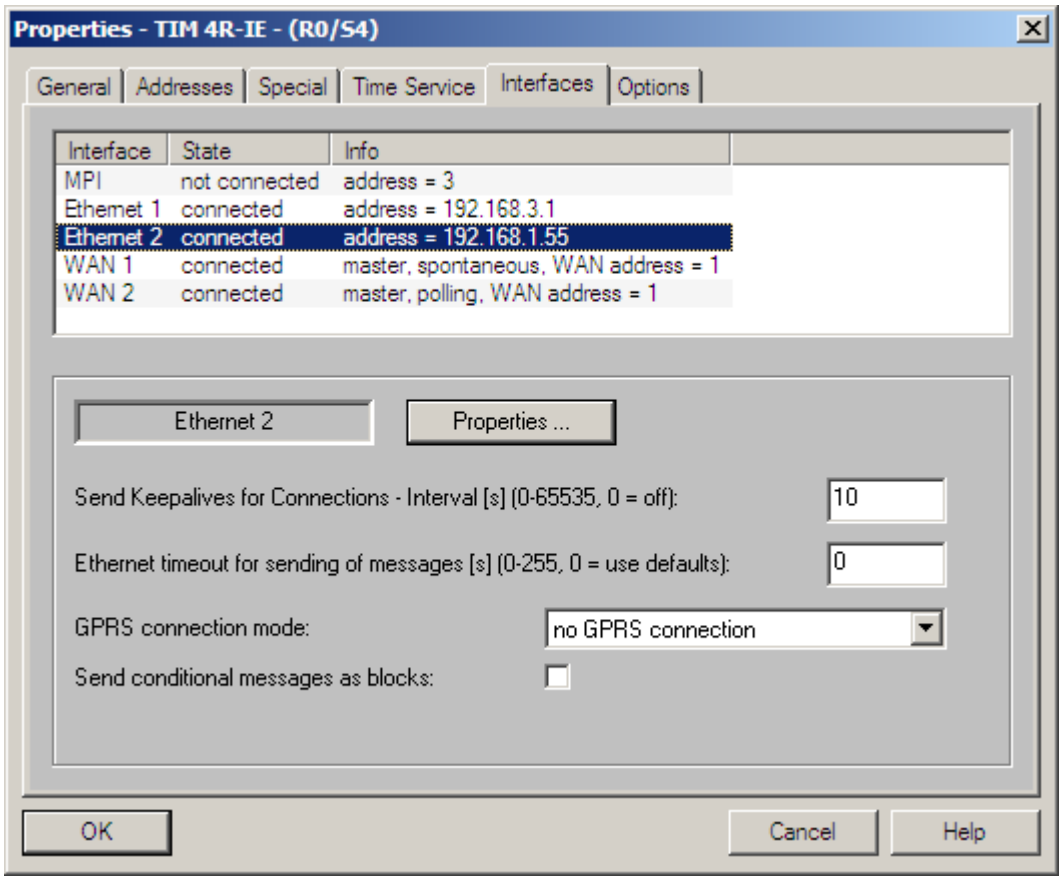


Figure 1-14 *Properties - TIM* dialog, *Interfaces* tab for an Ethernet TIM

If you click on an interface, the parameter boxes for the interface are displayed below the list:

- MPI
  Ethernet TIMs do not have an MPI interface. If a TIM 4 is installed in an S7-300 as a CP, it is displayed in the list of the "MPI" interface. If you select this by clicking on it with the mouse, you can configure the internal station MPI address of the TIM using the "Properties" button.

Note

The MPI interface is displayed only when one of the following CPU types is inserted:

- All variants of the CPUs 312, 312C, 313C, 314 and 314C
- The CPUs 315-2 DP and 315F-2 DP

- WAN 1 / 2
  Parameter assignment as classic WAN interface:

  The parameters of the WAN interfaces correspond to those of the TIM modules of the type TIM 3/TIM 4. You will find a description in the Section "WAN Access" tab (Page 41).

- WAN 1
  Parameter assignment of the serial interface as virtual Ethernet interface for simple Internet communication via GPRS.
  An Ethernet TIM that you want to connect as an MSC station (or node station) to a GPRS network via the GSM MD720-3 modem is connected to an Ethernet network via its serial interface.

  With a TIM 4R-IE, only WAN 1 supports the function of a virtual Ethernet interface. When networking interfaces in NetPro, remember that the left interface of 2 interfaces of the same type is not necessarily interface 1.

  Follow the steps below to configure the interface:

  – Select the serial interface (WAN 1 with a TIM 4R-IE).

  – From the drop-down list in the lower part of the dialog, select the network type "Ethernet" and assign it to the WAN interface by clicking the "New" button.
    The drop-down list entry "Ethernet" is grayed out.

  – Open the properties dialog of the network node by clicking the "Properties" button, then connect the WAN network nodes with an Ethernet network and configure the IP parameters. Close the dialog with "OK".

    Back in the "Interfaces" tab, the MD720-3 is automatically assigned as "Modem type" and the "GPRS connection mode" is enabled. The WAN interface now has parameter settings for a virtual Ethernet interface.

  – As the "GPRS connection mode", select either "MSC station" or "MSC node station".

    The remaining MSC configuration is done in the SINAUT configuration tool. You will find an overview of MSC configuration in the section GPRS / Internet communication: Overview of configuration (Page 15).

- Ethernet 1 / 2
  The parameters of the Ethernet interface are displayed below the list.

  In the list box for the (static) Ethernet interface, "Ethernet 1" (or "Ethernet 2") is selected as default. You set the IP parameters of the Ethernet interface by clicking the *Properties...* button, see Setting parameters for Ethernet nodes (Page 62).

  Note

  The two Ethernet ports of a TIM 4R-IE are not designed as a switch, but are intended for connection to different networks. Operation in the same Ethernet network is not permitted.

  If this is ignored, it will not be possible to generate SDBs for the TIM. This is detected during the verification in SINAUT node management and signaled.

  The IP addresses of the two interfaces must therefore differ in at least one of the three leftmost decimal (separated by a period) numbers (applies to the usual subnet mask 255.255.255.0).

When an Ethernet interface is selected, you have the option of setting the following parameters:

- *Send Keepalives for Connections - Interval [s] (0-65535, 0 = off)*:

  This value specifies the interval in seconds at which keepalives are sent. If the value is set to 0, no keepalive messages are sent.

  For GPRS connections, a value of 120 seconds is recommended. The value can also be selected here depending on the period in which the "conditional spontaneous" frames stored on the TIM must be sent (see parameter *Send conditional messages as blocks*). The keepalive interval should always be shorter than the interval for "dead peer detection" (DPD) on the MD741-1 modem. On the MD741-1, the default DPD interval is 150 seconds.

- *Keepalive timeout [s] (0-255, 0 = default)*:

  This value specifies the monitoring time in seconds when sending a keepalive. The acknowledgment of the message just sent must arrive within the monitoring time defined here. If the value is set to 0, the internal TIM default value is used (1 second).

  In GPRS networks, a message is usually acknowledged within 1 to 2 seconds. This may take longer depending on the load on the GPRS network. Experience has shown that an *Ethernet timeout* of 10 seconds is practical in GPRS networks.

- *GPRS connection mode*

  You can either set parameters for the Ethernet interface for IP communication according to the S7 protocol ("neutral" setting) or for data transmission by GPRS. If you are connected to a GPRS network, you have the alternative of transmission using the GPRS modem MD741-1 or for simple Internet communication with the MSC protocol using the GPRS modem MD720-3.

  An Ethernet interface cannot operate at the same time with the S7 protocol and the MSC protocol.

  In contrast to a normal "flat" Ethernet network in which every connected subscriber can communicate with every other subscriber, there are only point-to-point connections between station and master in GPRS networks. A direct connection from station to station is not strictly possible. Messages must be forwarded in GPRS networks via the master.

  To allow this, an Ethernet TIM can be used in the master that handles the routing of data messages between stations. To be able to establish the correct route in the GPRS network during configuration of the SINAUT connections, you will need to assign one of the following options to the Ethernet interface of the TIM on the GPRS network:

  – *No GPRS connection*

    On the Ethernet interface of the TIM, there is normal TCP/IP communication with the ST7 protocol; in other words, no subscriber is connected via GPRS and the MSC protocol is not enabled.

  – *GPRS master*

    In the role of "GPRS master", the Ethernet interface represents the highest level in the GPRS network hierarchy. Messages from TIMs in the role of "GPRS station" or "GPRS node station" to other stations in the network can then be routed over this interface.

  – *GPRS node station*

    In the role of "GPRS node station", the Ethernet interface is subordinate to the "GPRS master". This setting is normally selected for a TIM located in a node station; in other words, in a station to which other stations are connected over a different network. Messages to be sent from this node station interface to other stations in the network are forwarded via the TIM with the interface role "GPRS master".

  – *GPRS station*

    In the role of "GPRS station", the Ethernet interface is subordinate to the "GPRS master". Messages to be sent from this station interface to other stations in the network are forwarded via the TIM with the interface role "GPRS master".

  You can still enable the MSC protocol. This makes simple Internet communication with GPRS via DSL/Internet or via a GPRS network possible. On the TIM 4R-IE, this function is supported only by Ethernet interface 1.

  With the "MSC master", "MSC node station" or "MSC station" options, you specify the direction of the data transmission in SINAUT networks.

  – *MSC master* (only TIM 4R-IE)

    You use this option for the interface of a TIM 4R-IE in the master or for the interface of a node station connected to a lower level network with stations.

  – *MSC node station*

    You use this option for the interface of an Ethernet TIM in a node station connected to the master via a higher level network.

– *MSC station*

  You use this option for the TIM 3V-IE in a station.

The remaining MSC configuration is done in the SINAUT configuration tool. You will find an overview of MSC configuration in the section GPRS / Internet communication: Overview of configuration (Page 15).

- *Send conditional messages as blocks*

  Data transmission over a GPRS network is subject to fees depending on the amount of data transmitted. To minimize costs, smaller data packets can be collected and transferred in larger blocks if these messages are assigned the "conditional" priority; in other words, they do not need to be sent immediately, refer to the SINAUT software TD7onTIM, section Basic parameters of the data objects (Page 138) or the description "SINAUT TD7 software package for the CPU" > "Data point typicals".

  If the *Send conditional messages as blocks* option is enabled, the TIM transmits "conditional" messages in the following situations:

  – When the collected messages reach or exceed a size of 202 bytes.
  – If an important message needs to be transmitted immediately, "conditional spontaneous" messages already in memory are transmitted along with it.
  – If the collected messages have not reached a size of 202 bytes, but the TCP/IP keepalive interval has elapsed, the stored messages are sent instead of the keepalive.
  – If messages are sent via an MSC connection, conditional spontaneous messages are collected until 1 300 bytes have been reached or until an unconditional spontaneous message is transmitted.

1.5.7 "WAN Access" tab

WAN Access tab

In the *WAN Access* tab, you can configure the following WAN interfaces of the TIM modules of types TIM 3 and TIM 4. The parameters correspond to the WAN parameters of the *Interfaces* tab for the Ethernet TIMs:

- The internal interface that is normally assigned to a modem installed on the module
- The external interface that must be set up by the user by selecting the required network type and clicking on the *New...* button

Behind each interface that is to be connected with a network, there must be a network node of the corresponding type and this can be recognized by the *Properties* button being available for selection.

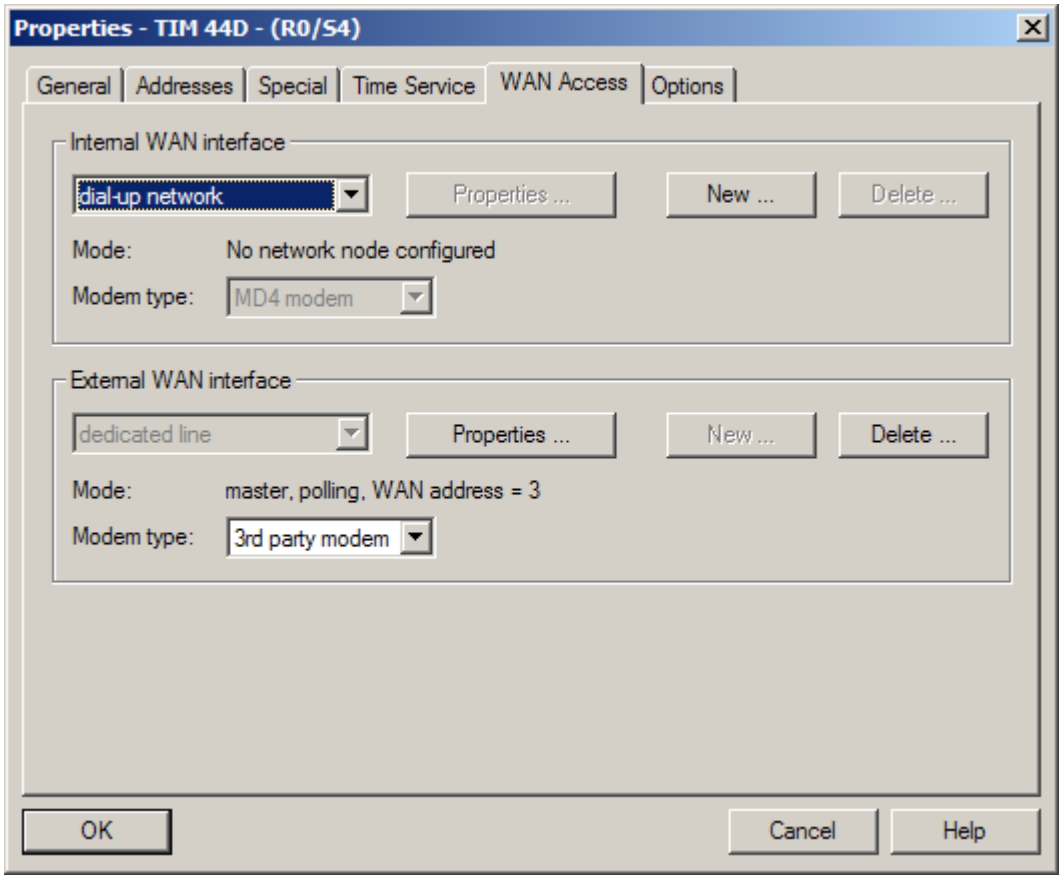


Figure 1-15 *Properties - TIM* dialog, *WAN Access* tab

For each of the two WAN accesses, you can make the following entries depending on the display in the *internal/external WAN interface*:

- Interface list box disabled with *Dedicated line*, *spontaneous network* or *dial-up network*:
  There is already a network node for this interface. Using the *Properties...* button, you can branch to the *Properties* dialog of the network node to make parameter settings. With the *Delete...* button, you can remove the network node.
  The *Properties* dialog of the interface is written when you set parameters for the network node.
- Interface list box selectable when *dedicated line*, *spontaneous network* or *dial-up network* is displayed:
  There is not yet a network node for this interface. To create a network node, you must select the corresponding interface type in the *Internal/external WAN interface* list box. With the *New...* button, you create a network node of the type displayed in the interface list box.
- Interface list box disabled with the display *Not available*:
  This interface cannot be operated by the current TIM module, no further parameter assignment possible.

The type of connector modem is displayed for each interface *Modem type* If a modem exists on the TIM module or can be connected externally. This type cannot be changed for the

internal interface. The type of a modem on the external interface must be configured over the list box.

The modem type is checked during the plausibility checks to establish whether or not it is compatible with the current network parameters. In addition to this, a default AT string for a SINAUT dial-up modem or GSM module is derived from the mode type and the network parameters *Baud rate*, *Message format* and *Connection type* (duplex, half duplex).

1.5.8 "Options" tab

Options tab

In the *Options* tab, you can set parameters for the message memory.

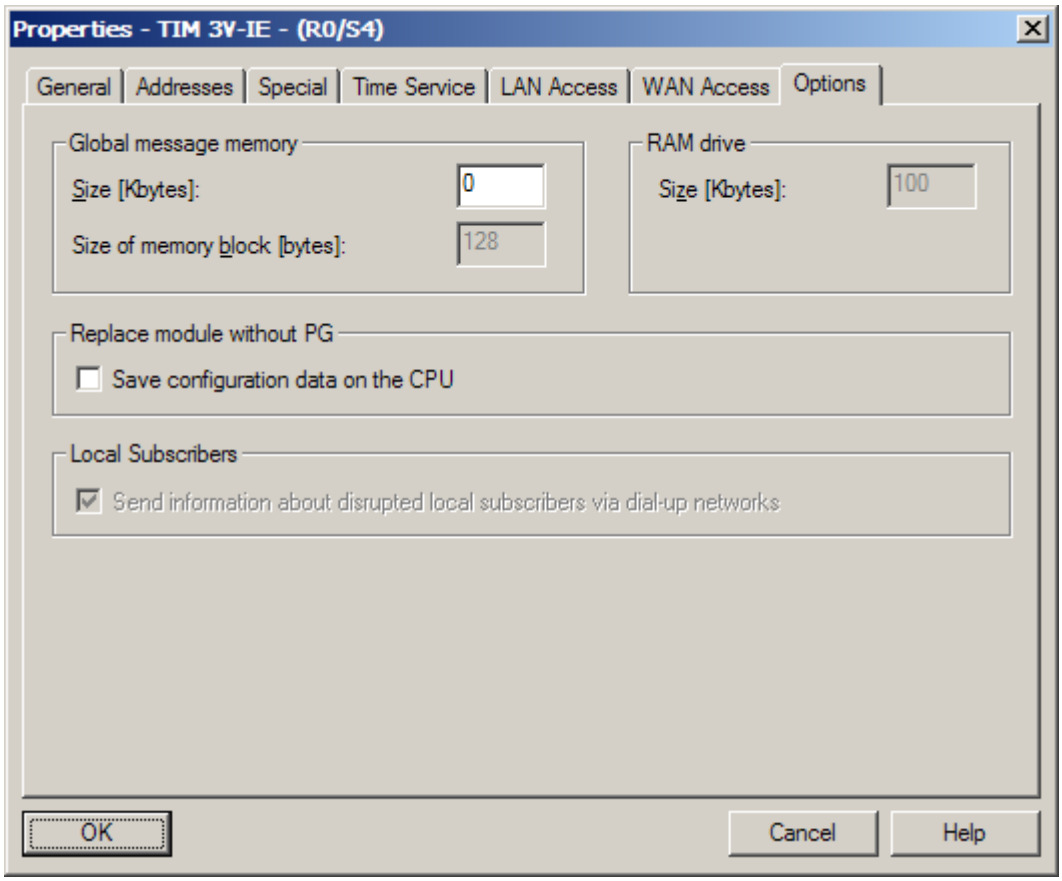


Figure 1-16 *Properties - TIM* dialog, *Options* tab

The following parameters can be set in the *Global message memory* area:

- *Size*:
  This is the size of the memory in which the messages to be transmitted are stored for all configured WAN drivers. If the value 0 is entered here, the entire free memory following startup is used.

Range of values: 0 ... 1024 Kbytes
Default: 0

- *Size of memory block*:
  This is the size of the blocks into which the global message memory is segmented. Each message to be transmitted occupies at least this space in memory. The size should be matched up with the size of the messages most commonly transmitted.
  If the size is set too small, longer messages must be distributed over several blocks. If the size is set too large, memory space is wasted with many of the messages.
  Range of values: 48 ... 65535 bytes
  Default: 64
- *RAM drive*:
  In this input box, you set the size of the RAM drive.
  The RAM drive is a restricted area in the main memory of the TIM module that can be created for special test purposes. The memory cannot be used for normal operation and is deleted again when the TIM module is restarted.
  On the TIM 3V-IE, the RAM drive is already set up at 100 Kbytes, the value 0 is displayed here and cannot be modified.
  On the TIM 3V-IE Advanced and the TIM 4R-IE, the RAM drive (100 Kbytes) can be modified.
  Range of values: 0 .. 1024 Kbytes
  Default: 0

In the *Replace module without PG* box, you can enable the following option for the Ethernet TIMs:

- *Save configuration data on the CPU*: (Ethernet TIMs only):
  If you enable this option, the system data blocks (SDBs) of the TIM module are stored on the CPU. If the TIM module fails, the defective TIM can be replaced by a TIM of the same type without leading to download the SDBs to the TIM using a PG. The TIM module obtains its SDBs from its local CPU during startup.
  If the TIM is configured as a standalone TIM without a CPU in the rack, this function is not available.

Note

If there is no C-PLUG inserted in a TIM 4R-IE, the configuration data is stored in flash memory. If there is a C-PLUG inserted in the TIM 4R-IE, the configuration data is stored automatically on the C-PLUG when it is downloaded. If you replace the module, you can insert the C-PLUG with the configuration data in the new module.

In the *Local Subscribers* box, you can enable the following option for the TIM 4R-IE:

- *Send information about disrupted local subscribers vial dial-up networks*:
  If this option is enabled, the TIM signals the failure of local subscribers over connected dial-up networks to the substations.

  To reduce costs resulting from the automatic connection establishment in dial-up networks when subscribers drop out, this option can be disabled.

1.5.9 Consistency check and saving

Saving and consistency check

Once you have completed the parameter assignment in the hardware configuration, the current version must be saved with the *Station / Save* menu. You can generate system data blocks (SDBs) using the *Save and compile* menu later since there are still other configuration steps necessary before the SDBs can be generated completely. When you close hardware configuration, a dialog opens automatically prompting you to save your entries.

When you select the *Save and compile...* function, a consistency check is run and a message is displayed if configuration errors are detected.

Note

To be able to acquire all the configured parameters for the TD7 software following changes in the hardware configuration of existing SINAUT installations when the system data is generated, the SINAUT configuration tool must first be started with connection configuration and then with subscriber administration. The project should be saved there, and the SDBs should be generated exclusively in *subscriber administration*.

Printing module information

The information on all configured modules in the current rack or for a selected module can be printed out using the *Station / Print...* menu.

1.6 Configuring networks and network nodes in STEP 7 / NetPro

For the following configuration steps,

- Configuring SINAUT connections
- Configuring SINAUT subscriber data
- Configuring SINAUT objects of an Ethernet TIM
- Generating system data blocks (SDBs) and data blocks (DBs)

a fully configured network is required.

During the initial network configuration, the following tasks are performed:

- The modules with network capability are connected to the networks
- A graphic view of the network consisting of one or more subnets is created
- The required properties and parameters for each subnet and each networked module are specified
- The network configuration is documented

Starting the parameter assignment dialogs for networks and network nodes

If you double-click on a network or network node icon or select the *Object Properties* menu in the context menu (right-hand mouse button), the *Properties* dialog opens to allow you to set parameters. Here, you can connect modules with networking capability with the networks and adapt the properties of the relevant object to your requirements. All parameters have default settings that simplify parameter assignment.

First the network attachments are made.

1.6.1 Generating network attachments

To network a project, the communication-compliant modules (for example CPU or TIM) must be connected to suitable networks. The modules in the station icons in the project window of the network configuration include interface and network node icons displayed in different colors according to the network type.

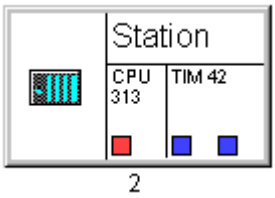


Figure 1-17 Icon of a master TIM station in the project window of the network configuration containing a TIM module and three network nodes

The station icon shows a master to containing a communication-compliant TIM 44D module. This module has three network nodes, visible as small squares in the module icon. If these network nodes are not connected to a network as in the example, the relevant network node is not networked.

You connect network nodes with the networks using the mouse by dragging the network node icons to the line of the required network.

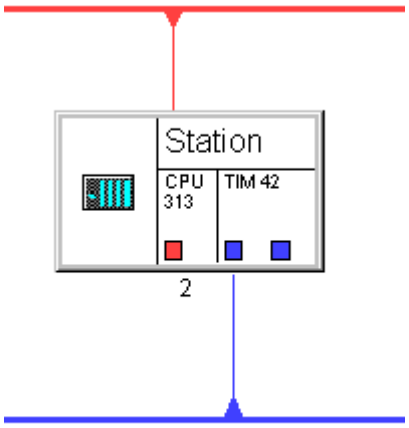


Figure 1-18 Station with three network nodes, two of which are networked

As an alternative you can attach to a network in the *Properties* dialog of the module available with the context menu (right-hand mouse button) and selecting the *Object Properties...* menu. The dialog is described in the section dealing with parameter assignment of network nodes.

With the simpler technique of dragging network node icons to the network line, only suitable partners can be networked; in other words an MPI node can only be connected to an MPI network. This makes incorrect attachment impossible.

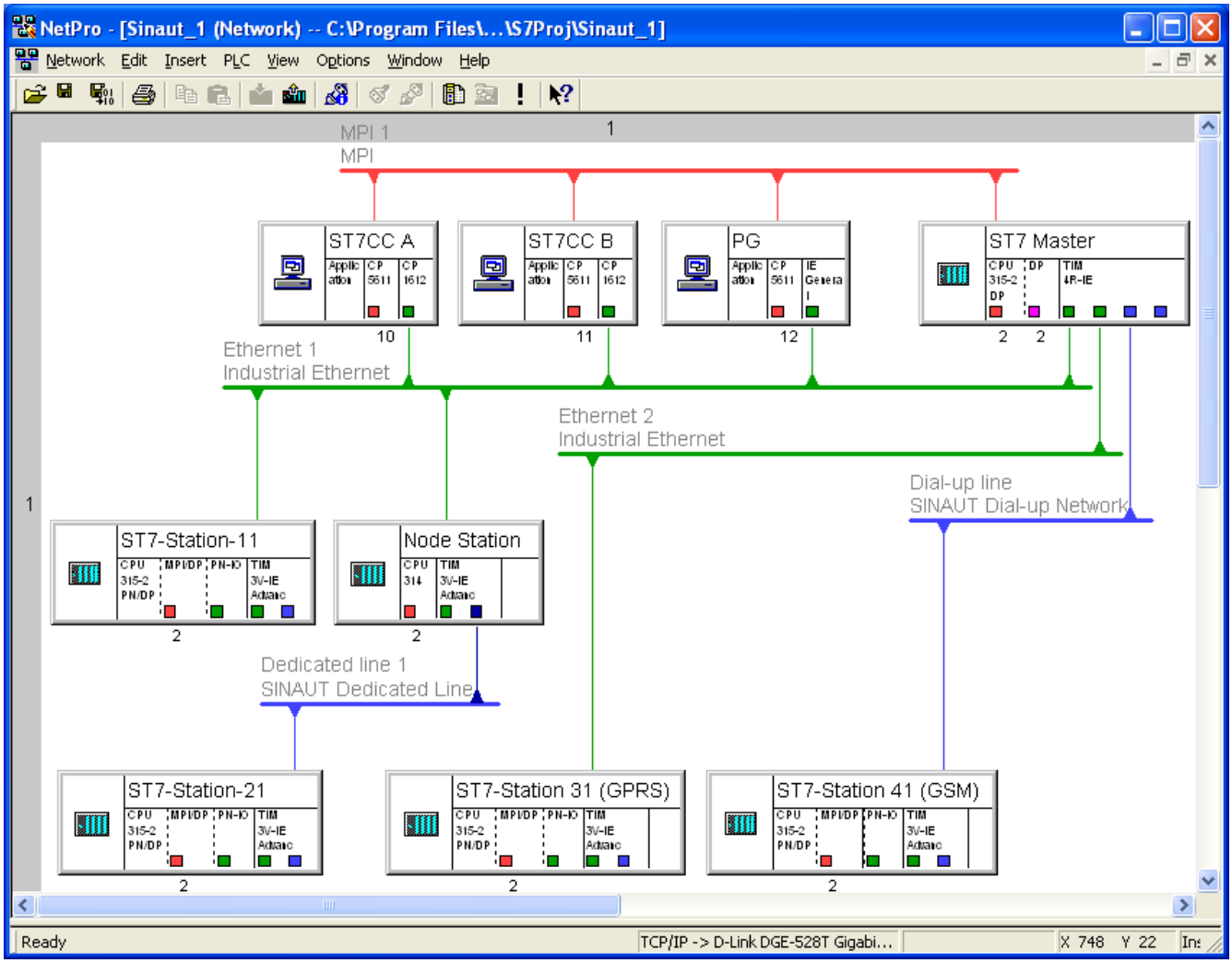


Figure 1-19 Networked sample project

The following network attachments were made in the sample project:

- The master station with the CPU 317-2 DP, the redundant ST7cc control center created as PC stations and the PG were connected to the MPI network and to Ethernet 1.
- The master TIM 4R-IE was further connected with Ethernet 2, dedicated line 2 and the dial-up network.
- The TIM 3V-IE in station 1 was connected to Ethernet 1.

- The TIM 3V-IE Advanced of the node station was connected to Ethernet 1 and dedicated line 1.
- The TIM 3V-IE in station 21 was connected to dedicated line 1.
- The TIM 3V-IE in station 31 (GPRS station) was configured for the MSC protocol and connected to Ethernet 2.
  Special feature: With this configuration, the WAN node to the right of the TIM is displayed green and is connected to the Ethernet network.
- The TIM 3V-IE Advanced in station 41 (GSM station) was connected to dial-up network 1.

The blue WAN connection of the master TIM and the connection of the node station to dedicated line 1 are shown in dark blue and indicate a connection in which the connected interface for the relevant subnet was configured as a master station. This is explained in the description of the network node parameter assignment.

Note

After changing connections, even if these are re-established again later, the SINAUT configuration tool with the *connection configuration* and the *subscriber administration* must be called.

Printing network information

The project can be printed and documented as a graphic or as text using the *Network / Print* menu.

1.6.2 Setting parameters for MPI networks

To set parameters for MPI networks, you open the *Properties - MPI* dialog by double-clicking on the MPI network, or using the *Object Properties...* context menu.

General tab

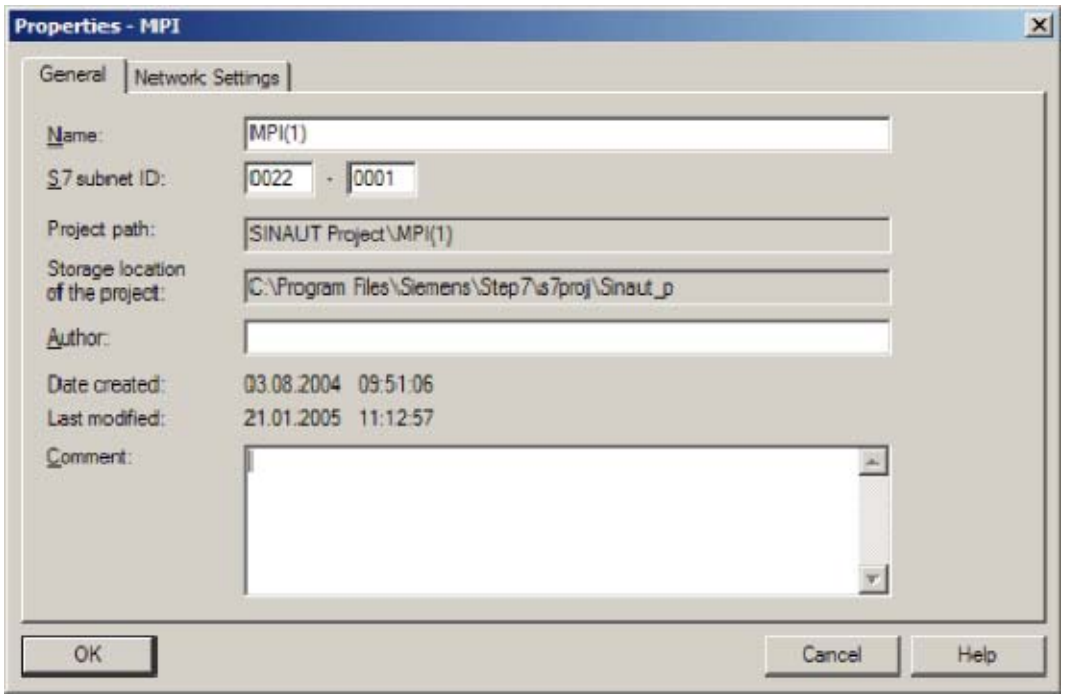


Figure 1-20 *Properties - MPI* dialog, *General* tab

The following parameters are available in this tab:

- *Name*:
  The default entry in the *Name* input box is the default name of the network. You can change this to suit your purposes. A new, modified name appears in the SIMATIC Manager and in the network configuration.
- *S7 subnet ID*:
  The subnet ID is made up of two numbers separated by a dash:
  - The number for the project
  - The number for the subnet

  If you want to go online with a PG without a consistent project, you must know the subnet ID. The subnet ID is also printed out when you print the network configuration.
- The *Project path* is displayed.
- The *Storage location of the project* is displayed.
- In the *Author* input box, you can enter the person who created the configuration.
- The *Date created* is displayed.
- The *Date of the last modification* is displayed.
- In the *Comment* input box, you can enter comments of up to 254 characters.

Network Settings tab

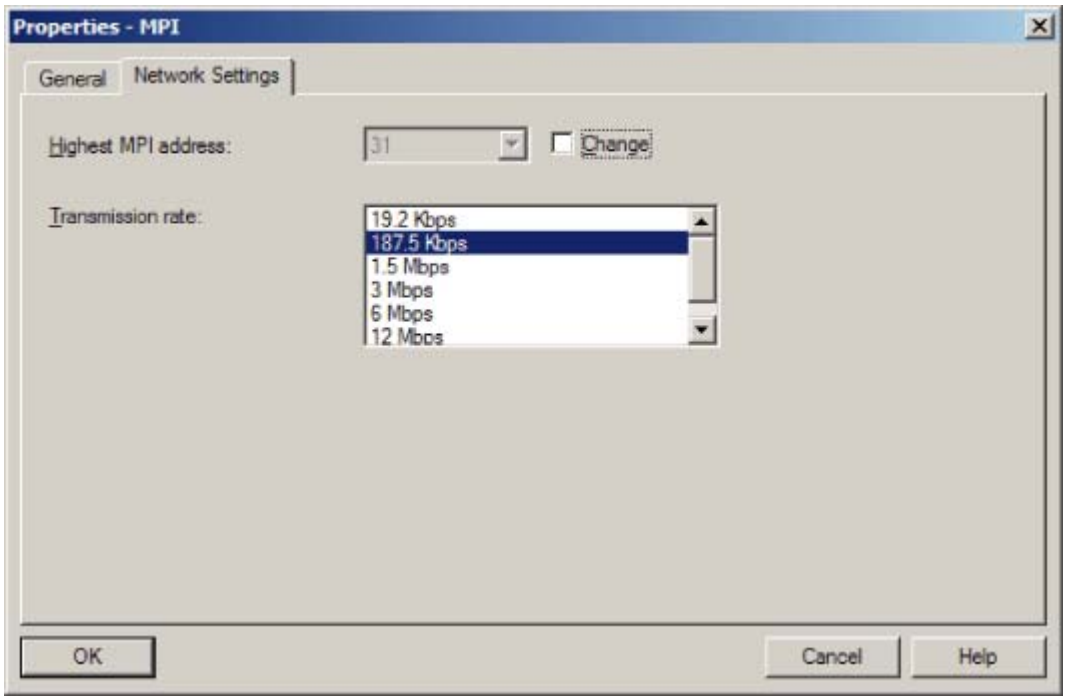


Figure 1-21 *Properties - MPI* dialog, *Network Settings* tab

The following parameters are available:

- The *Highest MPI address* is displayed.
  This is used to optimize the MPI network. It is advisable to retain the highest MPI address proposed by STEP 7.
- The *Change* option:
  By activating the option, you can modify the highest MPI address.
- *Transmission rate*:
  The setting of the transmission rate of the MPI network depends on the properties of the MPI subscribers used and must not be higher than the slowest subscriber. The default can normally be accepted.

1.6.3 Setting parameters for Industrial Ethernet the

General tab

You set the parameters for Industrial Ethernet in the *Properties - Industrial Ethernet* dialog.

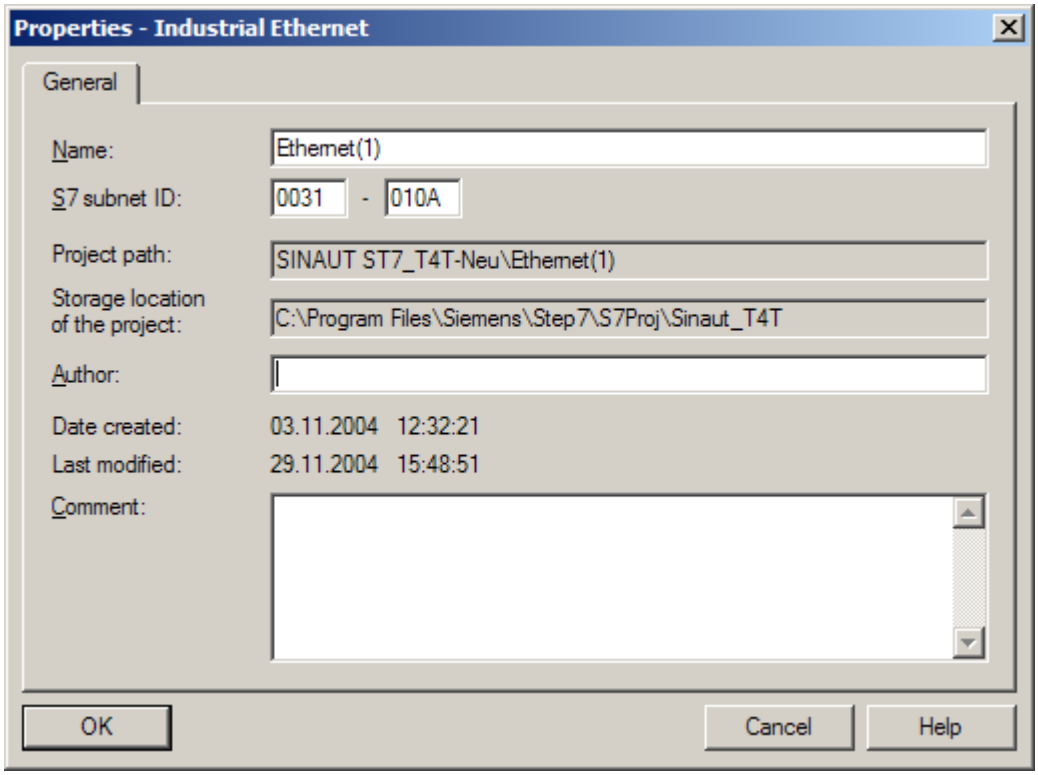


Figure 1-22 *Properties - Industrial Ethernet* dialog, *General* tab

The following parameters are available in this tab:

- *Name*:
  The default entry in the *Name* input box is the default name of the network. You can change this to suit your purposes. A new, modified name appears in the SIMATIC Manager and in the network configuration.
- *S7 subnet ID*:
  The subnet ID is made up of two numbers separated by a dash:
  - The number for the project
  - The number for the subnet

  If you want to go online with a PG without a consistent project, you must know the subnet ID. The subnet ID is also printed out when you print the network configuration.
- The *Project path* is displayed.
- The *Storage location of the project* is displayed.
- In the *Author* input box, you can enter the person who created the configuration.
- The *Date created* is displayed.

- The *Date of the last modification* is displayed.
- In the *Comment* input box, you can enter comments of up to 254 characters.

1.6.4 Setting parameters for SINAUT networks

You set parameters for SINAUT networks (WANs) in the *Properties - SINAUT Dedicated Line* or in the *Properties - SINAUT Dial-up Network* dialog. The parameters to be set in the following tabs always apply to the entire network and are identical for all attached network nodes or communication partners:

- *General*
  tab with general information and for modifying the module name or adding comments
- *Network settings*
  for setting the communication parameters of the current SINAUT network
- *Time Service*
  tab for setting parameters for time synchronization on the SINAUT network
- *Node List*
  tab with the list of all subscribers on the current SINAUT network
- *Time Slots* tab (only for the corresponding polling mode)
  to specify the time slots for polling

General tab

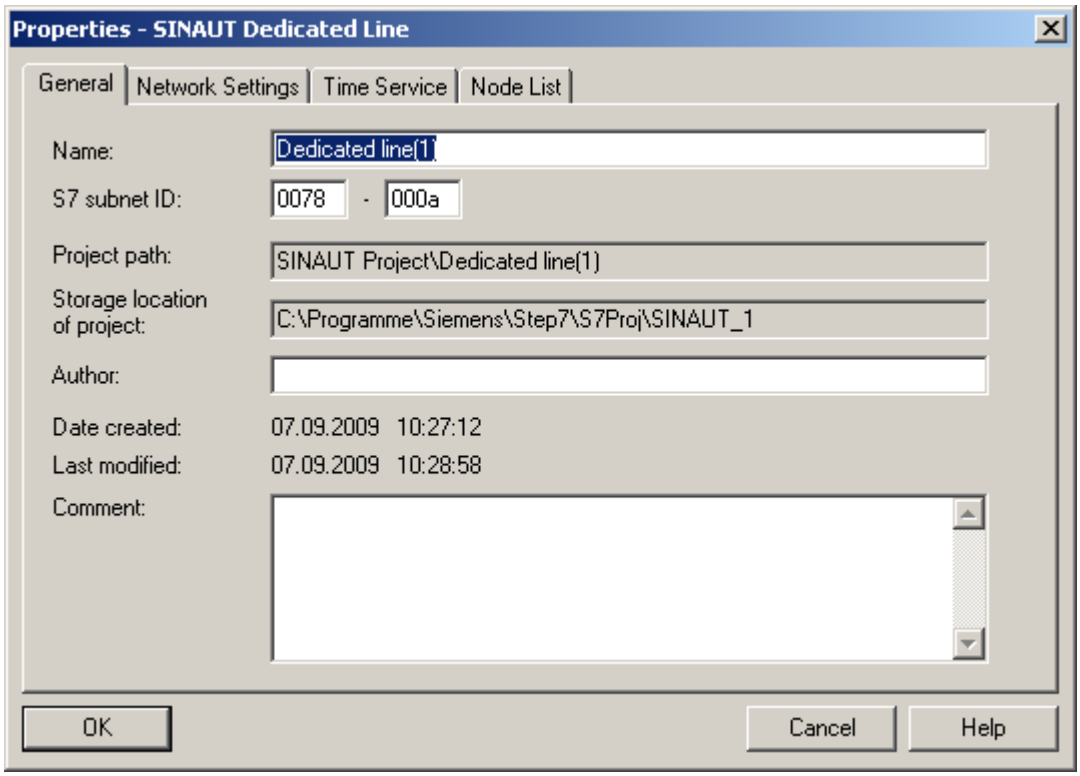


Figure 1-23 *Properties - SINAUT Dedicated Line* dialog, *General* tab

The following parameters are available in this tab:

- *Name*:
  The default entry in the *Name* input box is the default name of the network. You can change this to suit your purposes. A new, modified name appears in the SIMATIC Manager and in the network configuration.
- *S7 subnet ID*:
  The S7 subnet ID is made up of two numbers, one for the project and one for the subnet separated by a dash.
  If you want to go online with a PG without a consistent project, you must know the subnet ID. The subnet ID is also printed out when you print the network configuration.
- The *Project path* is displayed.
- The *Storage location of the project* is displayed.
- In the *Author* input box, you can enter the person who created the configuration.
- The *Date created* is displayed.
- The *Date of the last modification* is displayed.
- In the *Comment* input box, you can enter comments of up to 254 characters.

Network Settings tab

The *Network Settings* tab specifies the basic communication parameters for the current network.

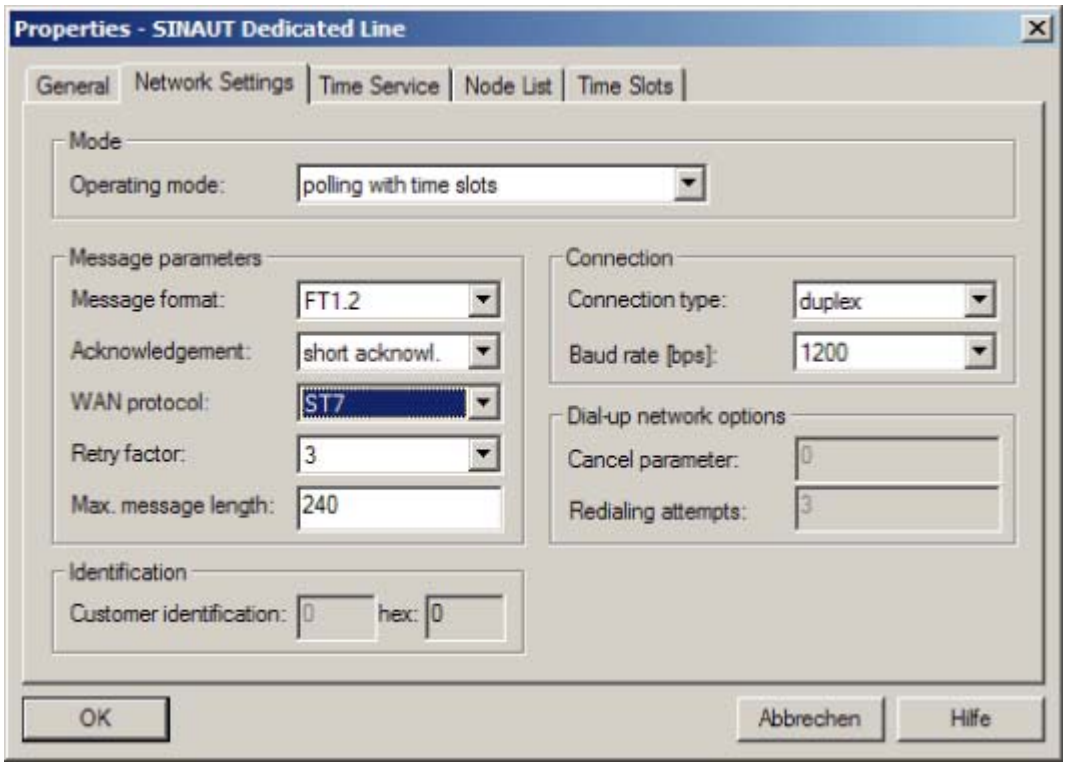


Figure 1-24 *Properties - SINAUT Dedicated Line* dialog, *Network Settings* tab

The following parameters are available:

- The *Operating mode* list box in the *Mode* area with the options:
  - Polling (dedicated lines only):
    In polling mode that is used with dedicated lines, the data exchange is is controlled by the master TIM. This polls the connected stations and node stations one after the other. Stations with data to transmit send it as soon as they are polled. Stations that do not currently have any data acknowledge the poll. Only data to be sent from the master TIM to the stations can be transferred at any time between two individual polls.
  - Polling with time slots (dedicated lines only):
    This mode differs from the polling mode because time slots are defined in which the polls take place. Each minute is divided into a selectable number of time slots. When using external wireless networks, the number of time slots per minute is generally prescribed by the relevant regulatory bodies. You configure the time slots to be used in the *Time Slots* tab.
  - Multi-master polling with time slots (dedicated lines are only):
    In this mode, the polls originate from several masters, once again in time slots. Here, various masters can be assigned different time slots for their polling.
    Multi-master polling with time slots is available only for ST7 networks with the FT2 message format with long acknowledgment. You configure the time slots in the *Time Slots* tab.
  - Spontaneous (dial-up networks only):
    Spontaneous mode is intended for data exchange in the public telephone network, the ISDN network, or the GSM network. Only the TIM with an important data change transfers its data spontaneously and waits for an acknowledgment from the partner. Prior to the actual data transfer, the TIM must first establish a dial-up connection to the partner. Following successful transmission of the data, the TIM waits for the acknowledgment. Following this, the dial-up connection is terminated immediately again if the partner does not use the existing connection to transfer any existing data.

In the *Message parameters* area, you can set the following parameters:

- *Message format*:
  The message format corresponds to IEC 870-5-1. The selection FT1.2 or FT2 depends on the modem. The standard modems MD2 - MD4 can handle both message formats, the GSM modules M20, TC35 and MC45 only FT2.
  Default: FT1.2
  - FT1.2 (8E1):
    Character format 8 data bits,
    even parity, 1 start bit, 1 stop bit
    Modem setting: Data format 11 bits
  - FT2 (8N1):
    Character format 8 data bits,
    no parity, 1 start bit, 1 stop bit
    Modem setting: Data format 10 bits

Note

For more detailed information on setting the message format, refer to the section on installing and putting a SINAUT modem into operation in the description of the displays and the connectors accessible from above.

- *Acknowledgment*:
  The type of an acknowledgment does not depend on the modem used. It is set dependent on the quality of the transmission line.
  Default: short acknowl.
  - short acknowl.:
    consists of one byte.
  - long acknowl.:
    consists of 5 bytes. The long acknowledgment is advisable in applications when interference produces spurious characters on the transmission line that could be interpreted by the TIM as a short acknowledgment.
    example: Bad wireless link
- *WAN protocol*:
  The WAN protocol is configured for the specific network.
  Default: ST7
  - ST7:
    The ST7 protocol is used as the WAN protocol. They should be the setting in all purely ST7 networks of an ST7 installation.
  - ST1:
    The ST1 protocol is used as the WAN protocol. This is required for communication in networks with old ST1 installations and in networks in which both ST7 subscribers and ST1 subscribers are connected.

  In a SINAUT dedicated line and dial-up network, either the ST1 or the ST7 protocol can be used. The selected network protocol applies to all subscribers connected to the relevant network or subnet.

  The following combinations of WAN protocol and mode with the dependent frame format are permitted:

Table 1- 1 Overview of the permitted WAN protocols with various modes (ST7 only)

Network type	Mode	WAN protocol	Asynchronous characters	Message format
Dedicated line / wireless network	Polling	ST7	11 or 10 bits	FT1.2 or FT2
Dedicated line / wireless network	Polling with time slots	ST7	11 or 10 bits	FT1.2 or FT2
Dedicated line / wireless network	Multi-master polling with time slots	ST7	10 bits	FT2
Dial-up network	Spontaneous	ST7	11 or 10 bits	FT1.2 or FT2

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

- *Retry factor*:
  This value decides how often a message that has not been acknowledged positively is repeated.
  Range of values: 0 .. 15
  Default value in dedicated lines and in the spontaneous network: 3
  Default in dial-up network: 7
- *Max. message length* (in bytes):
  The maximum message length is based on the longest ST7 message length within a network. Time values (for example sender retry time) for internal monitoring functions are derived from this information.

Range of values: 40 .. 240
Default: 240

Note

The default of 240 for the maximum message length should, whenever possible, not be changed since PG routing will not work with a maximum message length less than 240.

- *Connection type*:
  Range of values: Half duplex, duplex
  Default: Duplex
- *Baud rate*:
  This is the speed at which the TIM and modem communicate. The transmission rate in the current network is decided by the modem and is normally identical to that of the modem.
  If you want to operate the modem at a speed that is not one of the default speeds, set the next higher speed here (for example 19200 bauds with the modem operating and 14400 bauds).
  - Range of values on dedicated lines and in the spontaneous network:
    50, 100, 150, 200, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 bauds
  - Range of values in the dial-up network:
    1200, 2400, 4800, 9600, 19200, 38400 bauds
  - Default: 1200 bauds
- *Cancel parameter* (with dial-up networks only):
  This is the number of attempted dialing attempts until the attempt is finally aborted.
  Range of values: 0 ... 127
  Default: 0
  - Cancel parameter = 0:
    The call attempts are finally aborted when a connection was established 127 times in a row but no data could be transferred.
  - Cancel parameter = 1 ... 127:
    The call attempts are finally aborted when a connection was unsuccessful n times in a row, regardless of whether a connection could be established at all or whether data could not be transferred on an established connection.
- *Redialing attempts* (with dial-up networks only):
  This is the number of attempted calls until until a disruption is reported.
  Range of values: 1 ... 127
  Default: 3
- *Customer identification* (with dial-up networks only):
  The customer identification is used to specify whether connections can only be established to partners permitted for the network. The customer identification has the function of password protection in the relevant network.
  Range of values: 0 ... 65535
  Default: 0

  For more information on SINAUT ST1, refer to release 05/2007 of the manual.

Time Service tab

The *Time Service* tab specifies the extent to which time synchronization services will be executed by the master or in the case of a subnet by the node station in this network. Time synchronization for TIM modules is described in the tab of the same name in the properties dialog of the TIM (refer to the section Setting TIM module parameters (Page 24) "Setting TIM module parameters").

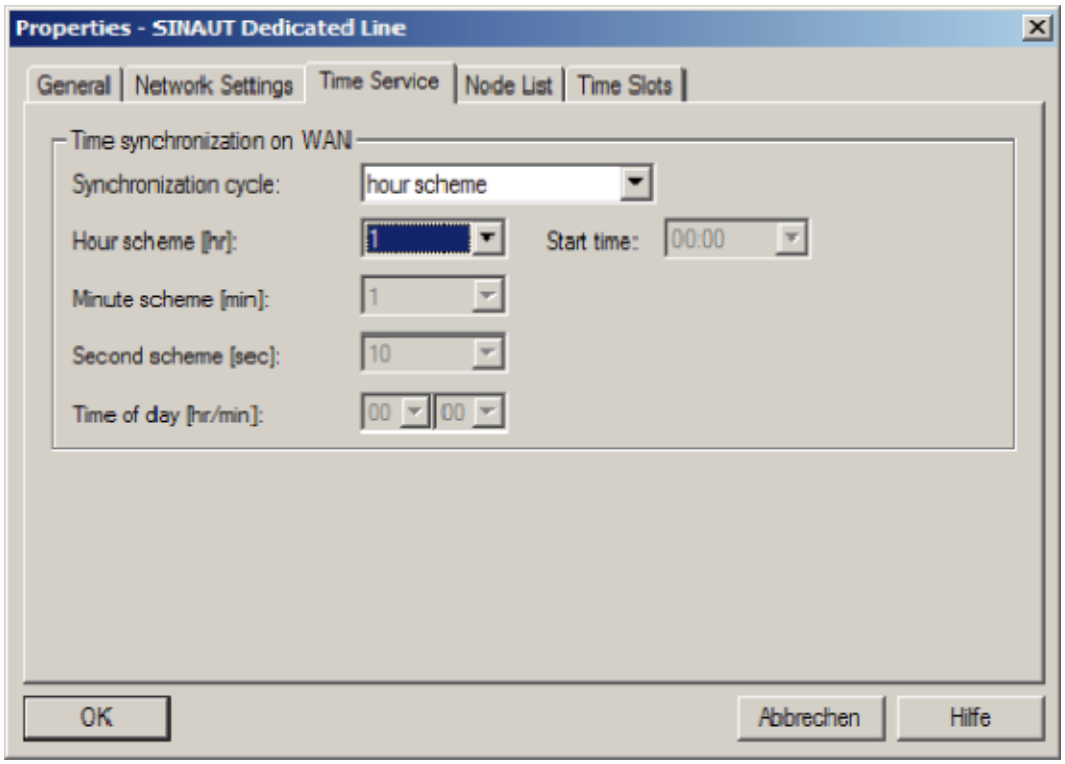


Figure 1-25 *Properties - SINAUT Dedicated Line* dialog, *Time Service* tab

You can set the following options for the *Synchronization cycle* parameter:

- No synchronization:
  There is no time synchronization on the relevant network.
- Hour scheme:
  The number of hours between synchronization activities can be set in the "Hour scheme" drop-down list box.
  - Start time:
    If the cycle for time synchronization is longer than 1 hour, you can set a start time for time synchronization in the "Start time" drop-down list box.
- Minute scheme:
  The number of minutes between synchronization activities can be set in the "Minute scheme" drop-down list box.

- Second scheme:
  The number of seconds between synchronization activities can be set in the "Second scheme" drop-down list box.
- Time of day:
  Synchronization takes place once a day. Set the time of day for the synchronization in the "Time of day" drop-down list box (for example 01:00).

On dedicated lines, a synchronization cycle of 1 hour is recommended and in dial-up networks, once a day, for example at 01:00 a.m.

In dial-up networks, time synchronization can also be used to check the availability of a subscriber and to fetch data.

Which TIM is synchronization master on this WAN is decided automatically based on the configuration of the network attachments in the properties dialog of the network node.

Node List tab

The node list displays all the communication subscribers connected to the current network; in other words, TIM modules. It also lists the station name, the WAN address and the configured node type making it easy to check these parameters throughout the network.

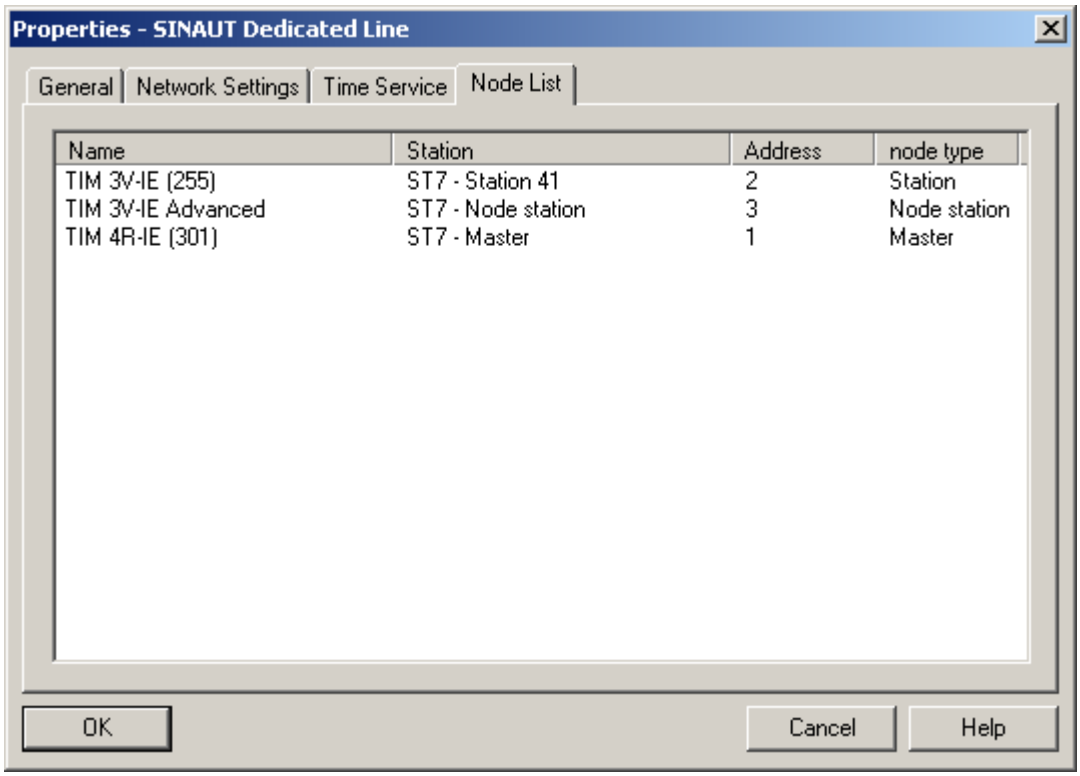


Figure 1-26 *Properties - SINAUT Dedicated Line* dialog, *Node List* tab

Note

If there is an asterisk after one or two addresses, the WAN address is not unique and must be changed.

Time Slots - Only in networks with a suitable polling mechanism

In the *Time Slots* tab, you can specify which time slots are used for transmission in the *Polling with time slots* and *Multi-master polling with time slots* modes.

To allow communication, at least one time slot must be selected since the master can only poll within the selected time range.

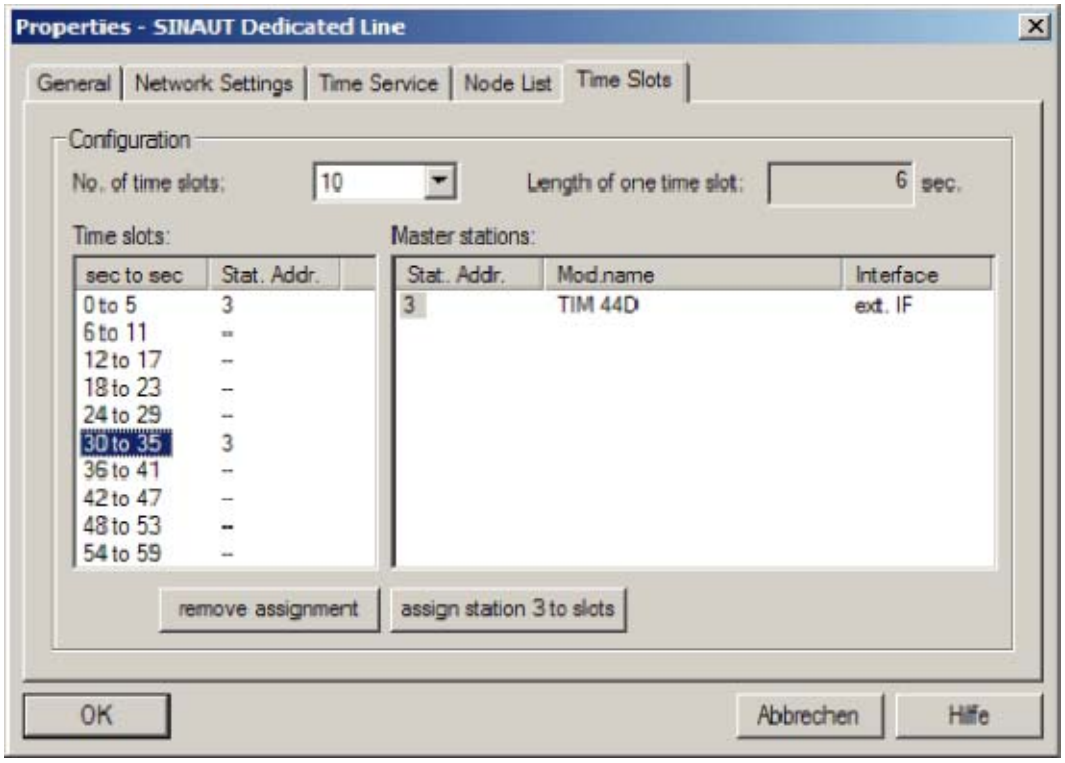


Figure 1-27 *Properties - SINAUT Dedicated Line* dialog, *Time Slots* tab

When setting the parameters for time slots, follow the steps below:

1. Specify the *No. of time slots* per minute. The *Length of one time slot* is calculated from this.
2. In the *Master stations* list, in the *Stat. Addr.* column, select the station address of the master station of the subnet.
3. In the *Time slots* list, in the *sec to sec* column, select the required time slot for calling this master station.
4. Click the *assign station # to slots* to assign the selected time slot to the selected master station.

Repeat this procedure if you want to use more than one time slot per minute.

In the multi-master polling with time slots mode, this must be done for each of the polling masters with different time slots.

You can delete a selected time slot again with the *remove assignment* button.

Note

Master TIM

The time slot method can now also be configured with the "SINAUT ST7 configuration software for the PG/PC" as of Version V4.2 for a master TIM without DCF7 receiver when a TIM 4R-IE with an Ethernet connection to an ST7cc-/ST7sc PC is used as the master TIM.

1.6.5 Setting parameters for MPI network nodes

General tab

The *General* tab informs you about general parameters of the MPI interface.

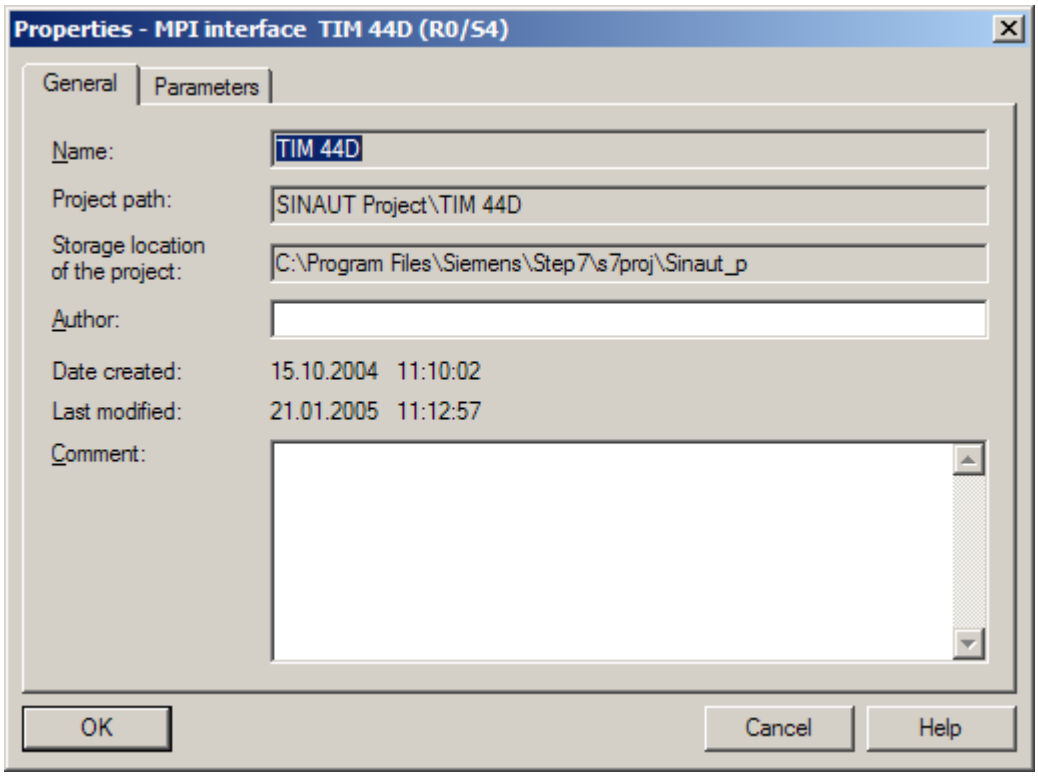


Figure 1-28 *Properties - MPI interface* dialog, *General* tab

The following parameters are available:

- Name:
  The *Name* box displays the name of the module in SIMATIC stations. You can only

change the default interface name in SIMATIC PC stations and other stations. A new, modified name appears in the SIMATIC Manager and in the network configuration.

- The *Project path* is displayed.
- The *Storage location of the project* is displayed.
- In the *Author* input box, you can enter the person who created the configuration.
- The *Date created* is displayed.
- The *Date of the last modification* is displayed.
- In the *Comment* input box, you can enter comments of up to 254 characters.

Parameters tab

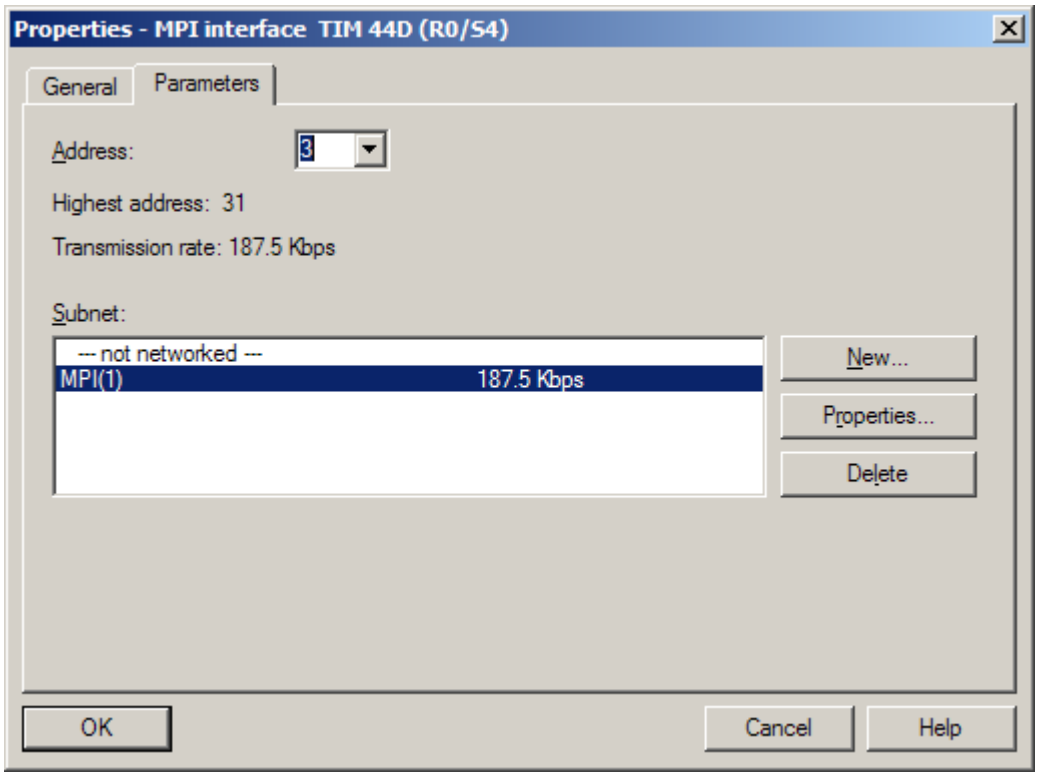


Figure 1-29 *Properties - MPI interface* dialog, *Parameters* tab

The *Parameters* tab provides the following parameters:

- The network *address* in the local MPI network.
  The address of the network node can be modified. There is a consistency check which blocks network addresses that have already been assigned.
- The *highest MPI address* in the network is displayed.

- The *transmission rate*:
  Just like the highest MPI address, this cannot be modified here but only in the parameter assignment of the MPI network.
- The *Subnet*:
  This lists all the networks of this type in the project. If the subscriber is not connected, the row *----not networked----* is shown as selected in the *Subnet* list.
  - If the current subscriber is connected, the row of the relevant network is shown as selected. You can set parameters for the current network with the *Properties* button.
  - If no connection exists, a network connection can be set up using the *New* button.
  - An existing connection can be deleted with the *Delete* button.

1.6.6 Setting parameters for Ethernet nodes

This type of interface is available among the SINAUT subscribers only for the Ethernet TIMs.

General tab

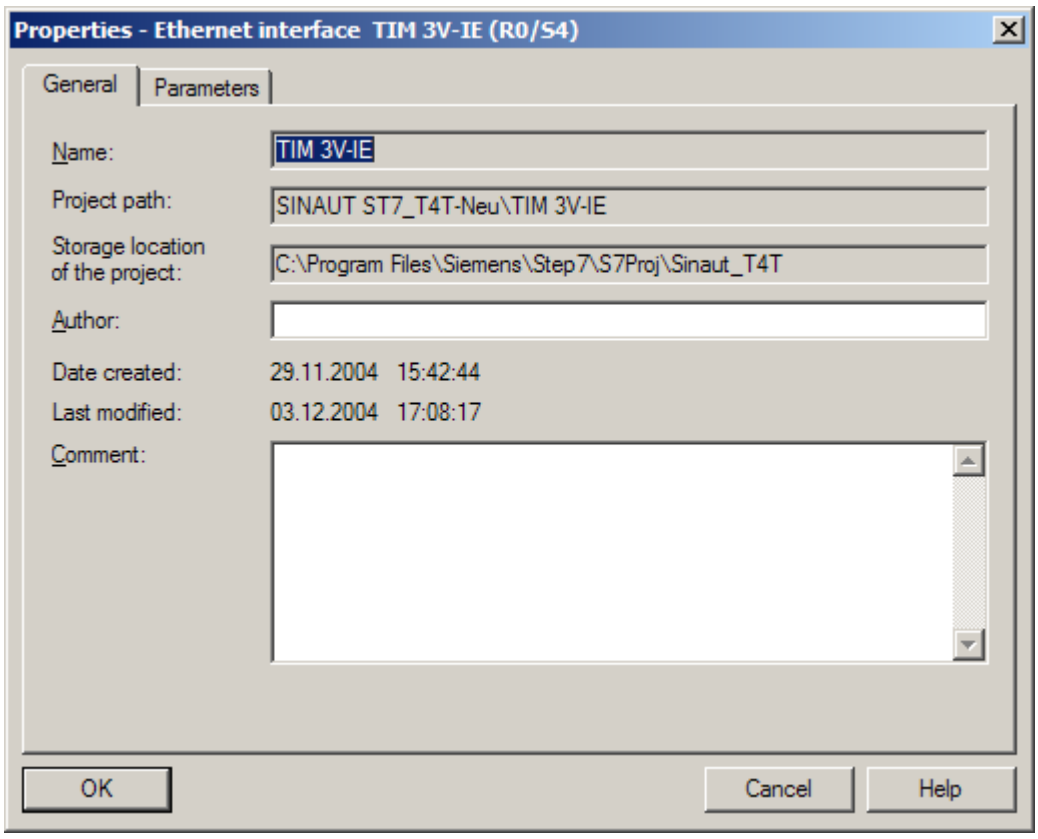


Figure 1-30 *Properties - Ethernet interface* dialog, *General* tab

The *General* tab informs you about general parameters of the Ethernet interface.

- Name:
  The *Name* box displays the name of the module.
- The *Project path* is displayed.
- The *Storage location of the project* is displayed.
- In the *Author* input box, you can enter the person who created the configuration.
- The *Date created* is displayed.
- The *Date of the last modification* is displayed.
- In the *Comment* input box, you can enter comments of up to 254 characters.

Parameters tab

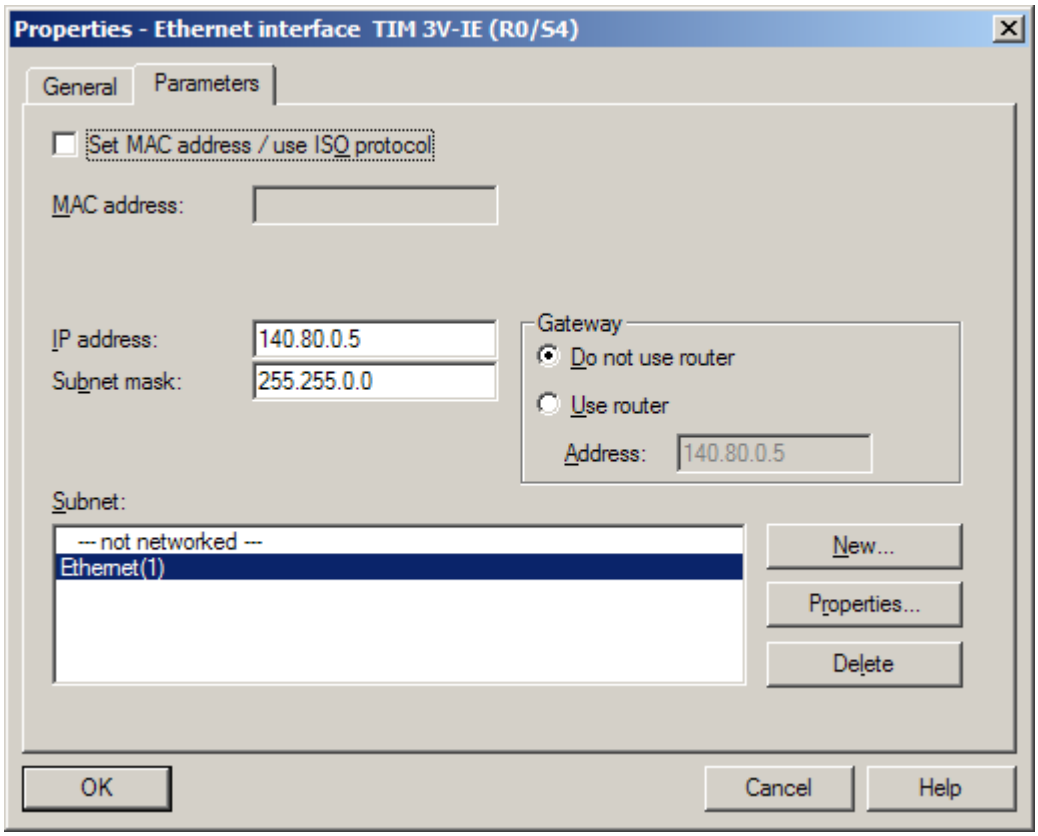


Figure 1-31 *Properties - Ethernet interface* dialog, *Parameters* tab

The *Parameters* tab provides the following parameters:

- *Set MAC address / use ISO protocol*:
  Since an Ethernet TIM does not use the ISO protocol but TCP/IP, this option remains disabled.
- The *IP address*:
  This cannot be changed here.

- The *Subnet mask*:
  This has the default value 255.255.0.0 and cannot be changed, in other words restricted here.
- In the *Gateway* box, you have the option of specifying whether data transmission is over a router.
  If a router exists, the IP address of the router is entered in the *Address* box.
- The *Subnet*:
  This lists all the networks of this type in the project. If the subscriber is not connected, the row *----not networked----* is shown as selected in the *Subnet* list.
  - If the current subscriber is connected, the row of the relevant network is shown as selected. You can set parameters for the current network with the *Properties* button.
  - If no connection exists, a network connection can be set up using the *New* button.
  - An existing connection can be deleted with the *Delete* button.

1.6.7 Setting parameters for WAN network nodes

You set parameters for SINAUT WAN networks in the *Properties - SINAUT Dedicated Line* or in the *Properties - SINAUT Dial-up Network* dialog in the following tabs:

- *General*
  tab with general information on the network node and entry of comments
- *Network Connection*
  tab for setting the most important network properties
- *Basic Param.*
  tab for setting the basic communication parameters
- *Dedicated Line*
  tab with parameters specifically for dedicated lines
- *Dial-up Network*
  tab with parameters specifically for dial-up networks
- *Call Parameters*
  tab with parameters specifically for call numbers
- *AT Initialization*
  tab for setting special AT strings when they are required

The relevant tabs are displayed depending on the network type.

Note

When setting parameters for network nodes, only the parameters that are practicable for the particular combination can be modified. This depends on the following:

- Network node type specified in the *Properties Network Node* dialog, *Network Connection* tab
  and the
- Operating mode specified in the *Properties Network* dialog, *Network Settings* tab.

General tab

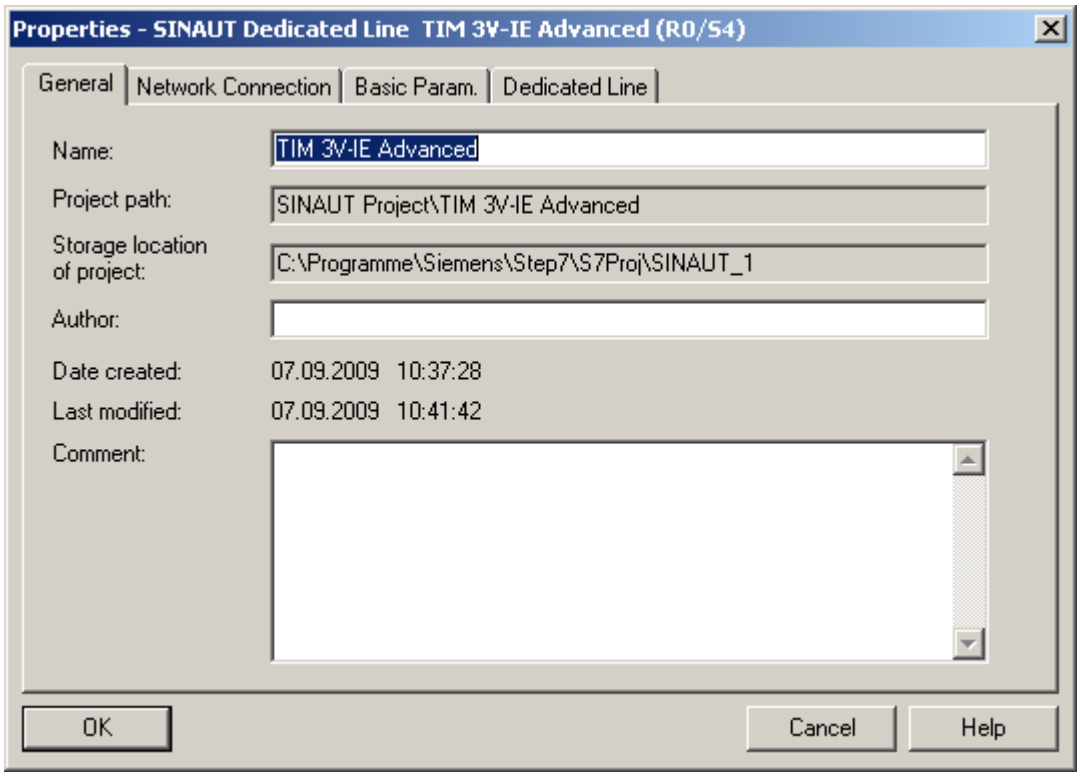


Figure 1-32 *Properties - SINAUT Dedicated Line TIM* dialog, *General* tab

The following parameters are available in the *General* tab:

- The *Name* box displays the name of the module in SIMATIC stations. You can only change the default interface name in SIMATIC PC stations and other stations. A new, modified name appears in the SIMATIC Manager and in the network configuration.
- The *Project path* is displayed.
- The *Storage location of the project* is displayed.
- In the *Author* input box, you can enter the person who created the configuration.
- The *Date created* is displayed.
- The *Date of the last modification* is displayed.
- In the *Comment* input box, you can enter comments of up to 254 characters.

Network Connection tab

The *Network Connection* tab allows you to set the most important networking properties of the WAN network node.

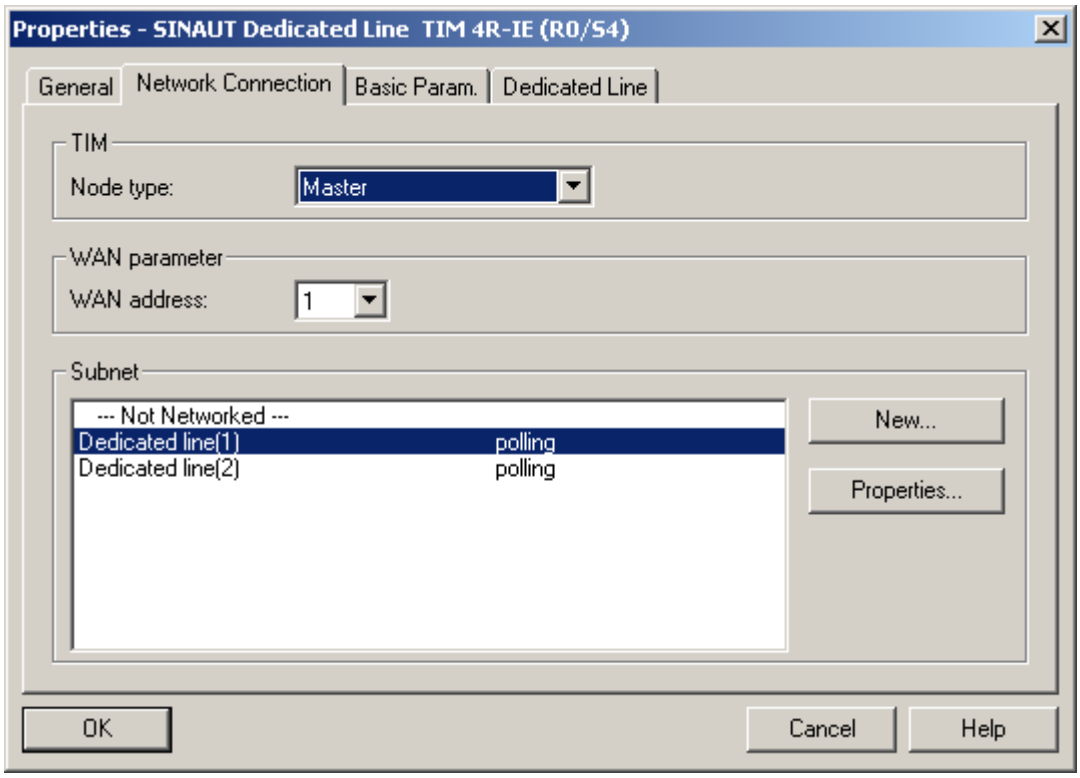


Figure 1-33 *Properties - SINAUT Dedicated Line TIM* dialog, *Network Connection* tab

The parameters here are:

- The *node type* selected in the list box:
  - The master station is the highest hierarchic level in the network. It generally collects information from the underlying network nodes and specifies settings for the nodes in the field.
  - A node station is at a hierarchically lower level than the master station or another node station and is at a higher level than one or more other stations.
  - A station is at a level close to the field and hierarchically below a master station or node station.
- The *WAN address* in the network.
  The unique WAN address of the node can be modified. A consistency check ensures that WAN addresses that have already been assigned cannot be selected.

  For more information on SINAUT ST1, refer to release 05/2007 of the manual.

- The *Subnet*:
  This lists all the networks of this type in the project. If the subscriber is not connected, the row *----not networked----* is shown as selected in the *Subnet* list.
  - If the current subscriber is connected, the row of the relevant network is shown as selected. You can set parameters for the current network with the *Properties* button.
  - If no connection exists, a network connection can be set up using the *New* button.
  - An existing connection can be deleted with the *Delete* button.

Basic Param. tab

The *Basic Param.* tab contains the communication parameters for the selected WAN node.

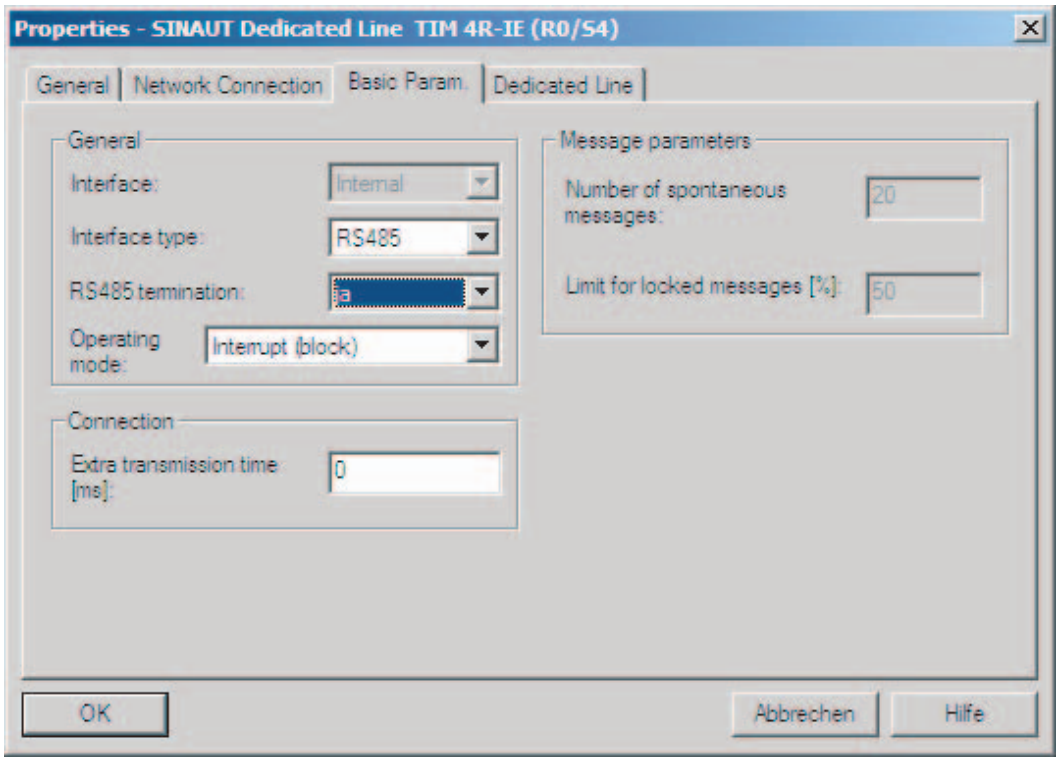


Figure 1-34 *Properties - SINAUT Dedicated Line TIM* dialog, *Basic Param.* tab

The following parameters are available:

- *Interface*:
  This output box indicates whether the current node is operated on the internal or the external WAN interface of the TIM.
- *Interface type* (TIM 4R-IE only):
  Here, you set the type of interface: RS-232 or RS-485 mode
- *RS-485 termination* (TIM 4R-IE only):
  Activation of the terminating resistor for the RS-485 bus

Note

If the interface is set to "RS-485", you will need to set the internal terminating resistor of the module for the RS-485 bus.

If the TIM 4R-IE is at the start of the RS-485 bus, which is normally the case, select the setting "yes". Otherwise select "no".

- *Operating mode*:
  This setting specifies whether the interface connected to the current node will be operated in interrupt or in DMA mode. Only one of the two interfaces of a TIM module may be operated in DMA mode.
  Range of values: Interrupt (block), DMA, Interrupt (single characters)
  Default: Interrupt (block)
  - Operating mode = Interrupt (block)
    This operating mode applies to the transmit and receive direction.
    The default mode Interrupt (block) is suitable for all connections. Four characters are transferred per block. Following this, there is an interrupt. The received characters are checked only after a complete message has arrived.
  - Operating mode = DMA
    This operation mode applies to the transmit and receive direction.
    The DMA mode should be used for connections with a high baud rate or heavy message traffic, however not for GSM networks.
    Only one of the two interfaces of a TIM module may be operated in DMA mode.
  - Operating mode = Interrupt (single characters)
    This operating mode is used only in the receive direction. In the transmit direction, the block mode continues to the used.
    This interrupt mode is suitable for extremely bad lines. An interrupt is triggered per transmitted character and each character this analyzes immediately after it is received allowing extremely good diagnostics of transmission errors. This mode is more reliable than the block mode but is slower.
- *Extra transm. time*:
  This is an offset added to the transmit retry time. The send retry time is calculated automatically on the TIM.
  From the *Extra transm. time* parameter, the character delay time can also be calculated (character delay time = extra transm. time divided by 5).
  An offset time should be entered in the *Extra transm. time* input box, for example when the send retry time cannot be calculated completely as is the case with satellite transmissions or wireless links over repeaters.
  Range of values: 0 .. 65535 ms
  Default: 0 (for M1 or M20 module: 400 ms)
- *Number of spontaneous messages*:
  This function is available only for the station and node station node types.
  Range of values: 0 .. 255
  Default for dedicated lines: 20
  Default for dial-up networks: 200
  - Number = 0 in polling mode:
    All spontaneous messages pending at the time of the first polling message are transferred.

– Number = 1 .. 255 in polling mode:
Maximum 1-255 spontaneous messages pending at the time of the first polling message are transferred.

In a dial-up station, the *Number of spontaneous messages* parameter decides after how many messages the master station has the opportunity of transferring its pending messages to the station.

- *Limit for locked messages*:
This parameter can only be set for dial-up networks and specifies the maximum percentage of locked messages in the send buffer. If this percentage is exceeded, the image method will be used for all new locked messages arriving. This prevents an overflow of the send buffer.
Messages are marked as locked if they can no longer be transferred to the addressees due to communication problems (known as the data brake).
Range of values: 0 ... 90% (If 0 is entered, the default setting is used)
Default: 50%

Dedicated Line tab

The *Dedicated Line* tab contains special parameters required only when using dedicated lines.

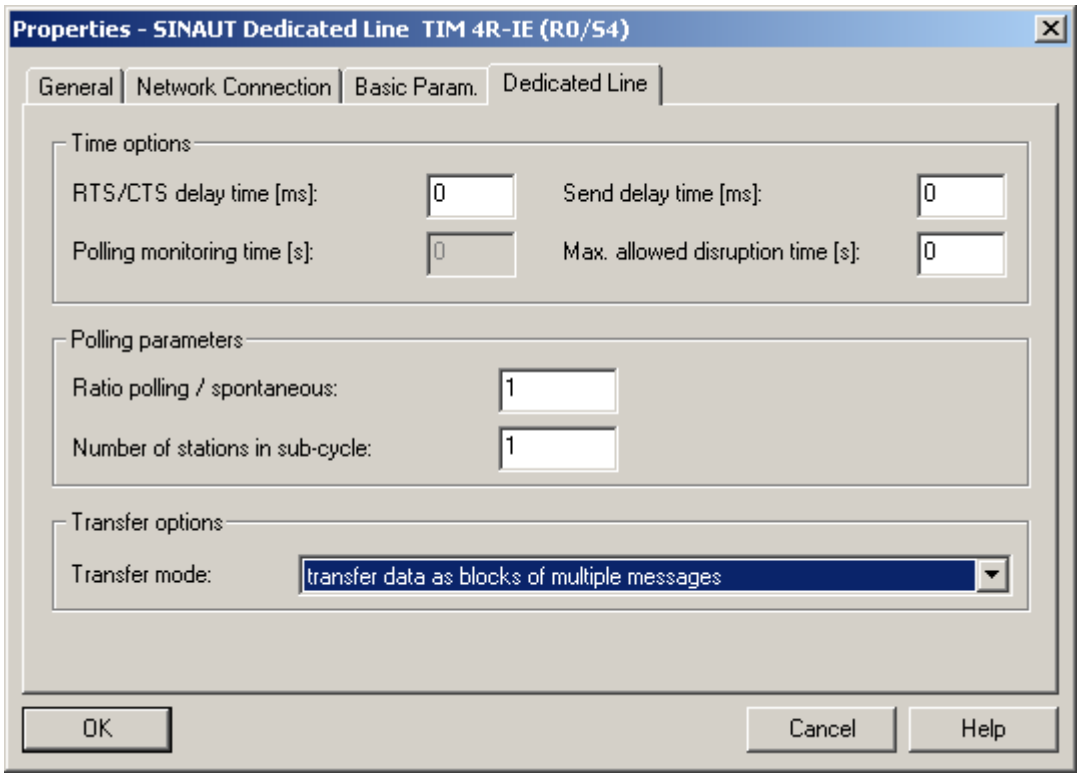


Figure 1-35 *Properties - SINAUT Dedicated Line TIM* dialog, *Dedicated Line* tab

The parameters for dedicated lines include:

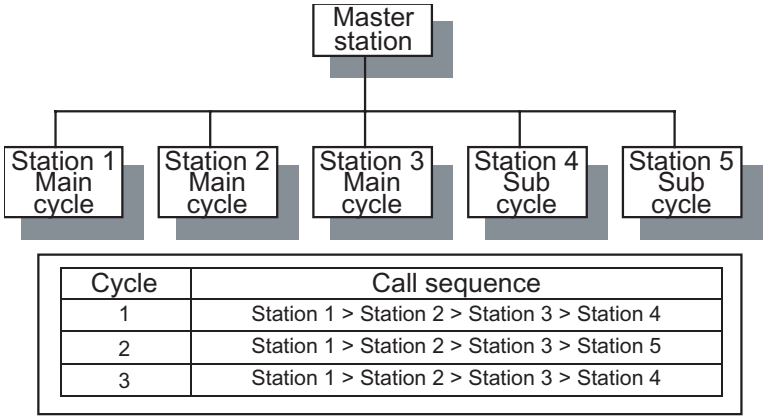
- *RTS/CTS delay time*:
  Setting the RTS/CTS delay time is required, for example when connecting a modem to the RS-485 port of the TIM module. The value is necessary for the RTS/CTS delay time can be found in the descriptions of the modems.
  Range of values: 0 .. 65535 ms
  Default: 0
  - RTS/CTS delay time = 0:
    After setting the RTS signal, transmission only starts when the CTS signal was set by the modem.
  - RTS/CTS delay time > 0:
    Transmission is not delayed until the CTS signal of the modem. After the RTS signal has been set, transmission is delayed for the selected time and then started immediately.
- *Polling monitoring time*:
  Specifies the latest time after which a station or node station TIM expects to be polled. If the TIM is not called after this time, it sends a message to its local CPU indicating that the master station is disrupted.
  Range of values: 0 .. 65535 s
  Default: 0 (0 means no monitoring)

  Even if no monitoring is set here, the TIM module registers the message traffic over the WAN and automatically sends the same fault message to its CPU if it does not register any message traffic for several seconds. If a maximum message length of 240 bytes is set, a with a retry factor of 3 and a transmission rate of 9600 bauds, the message is sent after approximately 4 seconds without message traffic, and at a transmission rate of 1200 bauds, after approximately 32 seconds.
- *Send delay time*:
  The send delay time is used only when the CTS signal comes from the modem (RTS/CTS time delay parameter = 0). As soon as the CTS signal comes from the modem, the send delay time is started. Data transmission is started only after this time elapses.
  This parameter is required, for example, when additional offset times are required to allow repeaters to start up on wireless links prior to starting data transmission.
  If 0 is entered, no send delay time is used.
  Range of values: 0 .. 65535 ms
  Default: 0
- *Max. allowed disruption time* (TIM 4R-IE only):
  Here, you can enter the tolerance time for a connection disruption detected by the TIM. If there is still a disruption on the connection when the set time has elapsed, the disruption is signaled to all connection partners of the disrupted station.
  Range of values: 0 ... 255 seconds
  Default: 0

  If disruptions occur frequently in networks (for example in some wireless networks), it may be helpful to increase the allowed disruption time without increasing the repetition factor for messages (see also properties dialog *Dedicated line*, *Network settings* tab). Increasing the allowed disruption time delays signaling of station failures and so reduces the number of organizational messages when stations return.
- *Ratio polling / spontaneous*:
  This output box displays the number of spontaneous messages that can be sent by a master station between two polls.
  Range of values: 0 .. 255
  Default: 1

- *Number of stations in sub-cycle*:
  This output box displays how many stations in the sub-cycle should be polled per main cycle.
  Range of values: 0 .. 250
  Default: 0

  The schematic shows a configuration with stations in the main and sub-cycle and the resulting polling order if 1 is set for the *Number of stations in sub-cycle* parameter.



Cycle	Call sequence
1	Station 1 > Station 2 > Station 3 > Station 4
2	Station 1 > Station 2 > Station 3 > Station 5
3	Station 1 > Station 2 > Station 3 > Station 4

Figure 1-36 Main cycle - sub-cycle

- *Transfer mode*:
  This parameter specifies the form in which ST7 data messages are sent when using the ST7 protocol.
  Range of values:
  - transfer data as blocks of single messages
  - transfer data as blocks of multiple messages
  Default: Single messages

Dial-up Network tab

The *Dial-up Network* tab contains special parameters required only when using dial-up networks lines.

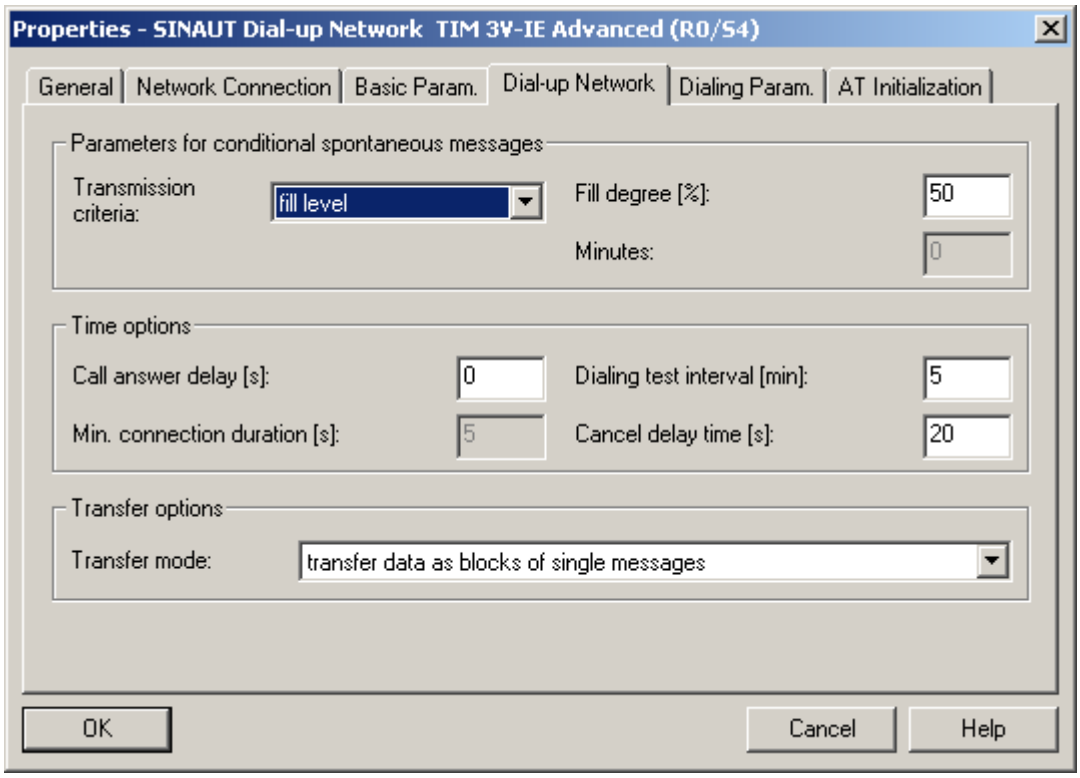


Figure 1-37 *Properties - SINAUT Dedicated Line TIM* dialog, *Dial-up Network* tab

The dial-up network parameters include:

- *Transmission criteria*:
  This setting controls connection establishment for the transmission of conditional spontaneous messages. The transmission criterion for conditional spontaneous messages can only be set for stations and node stations.
  Range of values: Standard conditions, fill level, time of day, time scheme
  Default: Standard conditions
  - Standard conditions:
    A connection is not established for existing conditional spontaneous messages. The conditional spontaneous messages are sent only when a connection is established for sending unconditional spontaneous messages or a buffer overflow is threatening or the connection is established by the other end.
  - Fill level:
    When the send buffer is filled to the specified level with conditional spontaneous messages, the TIM module automatically attempts to establish a connection and to transmit the messages.
    - Input box *%*: Entry of the send buffer fill level as a percentage (default: 50%)
  - Time of day:
    A connection is automatically established and the messages sent at the specified time of day. The time must be entered:

- Input box *Hours*: Entry of the time (hour)
- Input box *Minutes*: Entry of the time (minute)

– Time scheme:
A connection is automatically established and the messages sent at the specified time intervals. The interval must be entered:
- Input box *Hours*: Entry of the hour value for the send interval
- Input box *Minutes*: Entry of the minute value for the send interval

- *Call answer delay*:
This sets the time that the WAN driver waits before answering an incoming call. This allows time to answer a telephone call if a telephone is attached parallel with the TIM on a shared telephone connection. The value 0 means there is no call answer delay.
Range of values: 0 ... 60 s
Default: 0 s

- *Min. connection duration*:
Here, the minimum connection duration of a dial-up connection can be set. This may be required in fast dial-up networks to be able to wait for the response of subscribers during a GR before the connection is terminated.
Range of values: 0 ... 65535 s
Default: 5 s
0 means that there is no minimum connection duration.

- *Dialing test interval*:
This specifies the time in minutes for a test interval. A test interval is started when no connection to a particular subscriber could be established from a master TIM after the specified number of retries.
Following the test interval, the WAN driver automatically reattempts to establish a connection to the specified subscriber. If a connection cannot be established, the test interval starts again. If the test interval is running and the WAN driver gets a new message to be sent to the disrupted subscriber, it does not wait for the test interval to end but attempts to establish a connection immediately and to send the message.
Range of values: 0 ... 255 minutes
Default: 5 minutes

- *Cancel delay time*:
This parameter specifies how long a dial-up connection is retained when the send buffers of the TIM module are full and it can send no further messages or data to the CPU.
All messages received over the WAN interface are acknowledged negatively until this time has elapsed. Due to the negative acknowledgment of the previously sent message, the communication partner will repeat the message after the send retry time.
If the cancel delay time has elapsed, the connection is terminated.
Range of values: 0 ...255 seconds
Default: 0

- *Transfer mode*.:
This parameter specifies the form in which ST7 data messages are sent when using the ST7 protocol.
Range of values:
- transfer data as blocks of single messages
- transfer data as blocks of multiple messages
Default: Single messages

Dialing Param. tab

The *Dialing Param.* tab appears only with dial-up network nodes and includes all parameters specific to call-numbers.

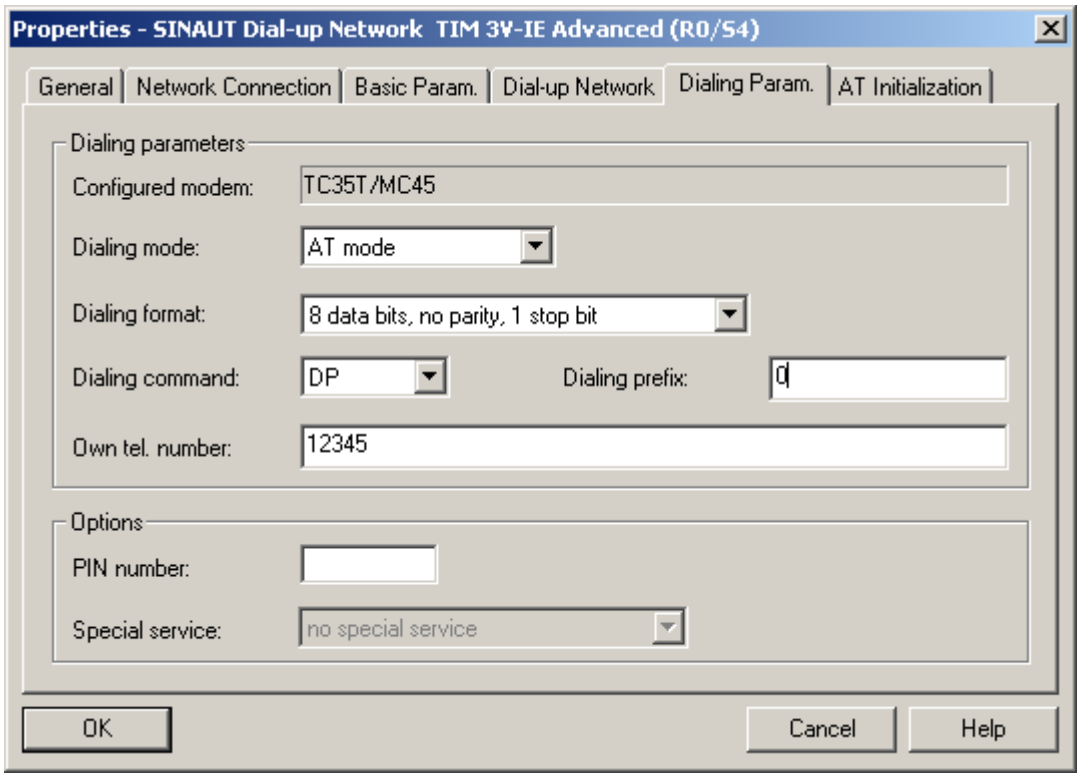


Figure 1-38 *Properties - SINAUT Dedicated Line TIM* dialog, *Dialing Param.* tab

The parameters here are:

- *Configured modem*:
  The modem specified in the hardware configuration is displayed.
- *Dialing mode*:
  Specifies how the modem is controlled. Note that only the AT mode can be used on the internal interface of the TIM. A choice between AT mode and V.25bis is possible only for the external WAN interface.
  Range of values: AT mode, V.25bis
  Default: AT mode
- *Dialing format*:
  The data format of the dial-up phase depends on the type of modem. The following settings are possible:
  8 data bits, no parity, 1 stop bit
  8 data bits, odd parity, 1 stop bit
  8 data bits, even parity, 1 stop bit
  8 data bits, no parity, 2 stop bits
  8 data bits, odd parity, 2 stop bits
  8 data bits, even parity, 2 stop bits
  7 data bits, no parity, 1 stop bit
  7 data bits, odd parity, 1 stop bit

7 data bits, even parity, 1 stop bit
7 data bits, no parity, 2 stop bits
7 data bits, odd parity, 2 stop bits
7 data bits, even parity, 2 stop bits

Default:
- in AT mode: 8 data bits, no parity, 1 stop bit
- with V.25bis: 7 data bits, even parity, 1 stop bit

Note

The data listed above applies only to the dialing phase. The data format in the data phase is set at switch 5 on the MD3 modem.

It is only necessary to set the dialing format on older modems that do not support "Autoband". With "Autoband", the modem can determine the character and data format in the dialing phase automatically based on the first AT string.

- *Dialing command*:
  This is the dialing command for the local modem. The following dialing commands are possible:
  - D (AT command)
  - DP (AT command, pulse dialing)
  - DT (AT command, tone dialing)
  - CRN (V.25bis)
  - CRNP (V.25bis, pulse dialing)
  - CRNT (V.25bis, tone dialing)

  Default: D. This default modem dialing command should be used where possible.
- *Dialing prefix*:
  This is the access number (outside line) for a private branch exchange (typical entry 0 or 9) or for an alternative telephone provider. A number up to 12 digits long can be specified. With direct connection to the dial-up network and without an alternative telephone provider, this parameter can remain empty.
  The dialing prefix can be changed again in the *Properties of subscriber* dialog.
- *Own tel. number*:
  Here, you enter your own telephone number for the network node including the area code. This telephone number can no longer be changed later in the *Properties of subscriber* dialog.

Note

In dial-up networks, in which another subscriber within the same local network cannot be dialed with the local area code, it is advisable to enter your own telephone number in the *Own tel. number* box without area code and to specify the area code in *Dialing prefix*.

- *PIN number*:
  Here, the PIN number for a GSM module must be entered so that this can be transferred from the TIM module to the module.

CAUTION
If an incorrect PIN is entered, the SIM card in the module might be disabled. If the fault LED lights up during connection establishment, the diagnostic buffer of the TIM must be checked because an entry for a bad PIN is generated here.

- *Special service*:
  The following SMS special services are available:
  - No special service

    The node does not use any SMS service.
  - SMS via fixed network (TAP)

    Sending of SMS messages to an SMS center that uses the TAP protocol
  - SMS via fixed network (UCP)

    Sending of SMS messages to an SMS center that uses the UCP protocol
  - SMS via GSM network

    Sending SMS messages to an SMS center with digital access

Note

If two MD3 modems communicate with each other, they must not be operated in the *1200 baud, half-duplex, AT mode* mode.

AT Initialization tab

The *AT Initialization* tab appears only in dial-up network nodes and when the *AT mode* is selected as *Dialing mode* in the *Dialing parameters* tab. The string stored here is formed automatically from the previously set dial-up network parameters and the SINAUT dial-up modem selected for the network node (MD3, MD4 or GSM modem TC35, MC45, MD720-1).

If "third-part modem" is set for the network node, no initialization string is displayed. You will then need to enter the correct string for the modem you are using. The initialization string set here is transferred to the modem operating in AT mode only when the TIM starts up.

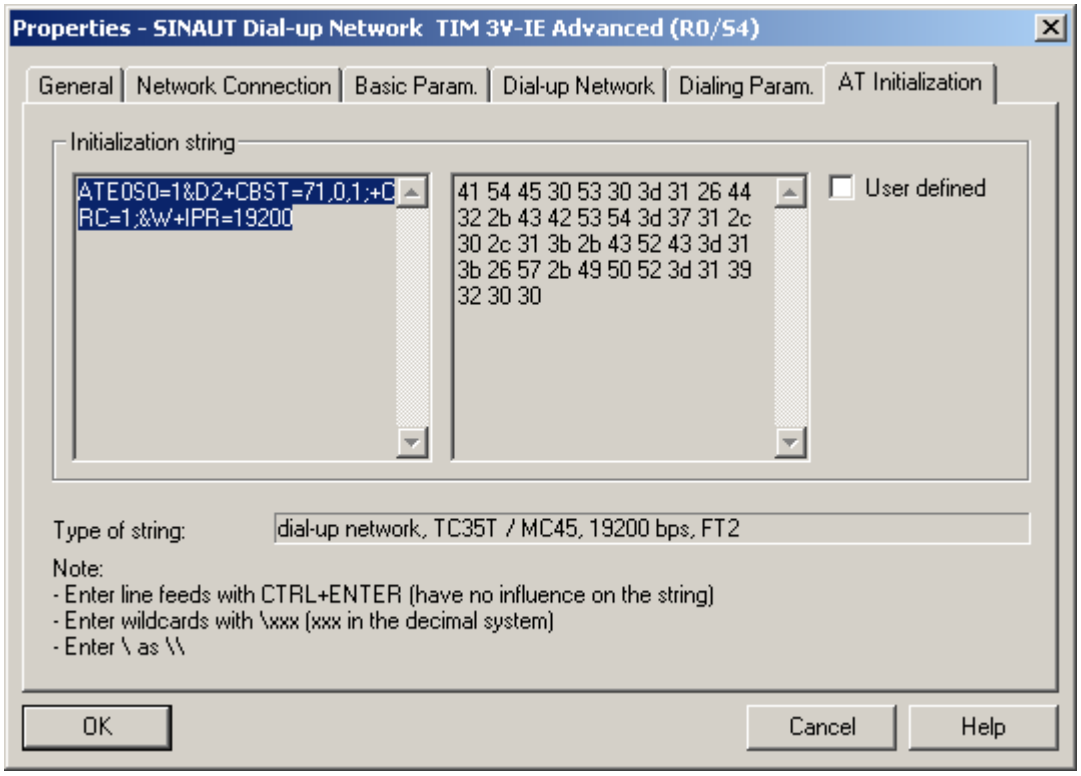

Properties - SINAUT Dial-up Network TIM 3V-IE Advanced (R0/S4)
General
Network Connection
Basic Param.
Dial-up Network
Dialing Param.
AT Initialization
Initialization string
ATE0S0=1&D2+CBST=71,0,1;+C
RC=1;&W+IPR=19200
41 54 45 30 53 30 3d 31 26 44
32 2b 43 42 53 54 3d 37 31 2c
30 2c 31 3b 2b 43 52 43 3d 31
3b 26 57 2b 49 50 52 3d 31 39
32 30 30
User defined
Type of string:
dial-up network, TC35T / MC45, 19200 bps, FT2
Note:
- Enter line feeds with CTRL+ENTER (have no influence on the string)
- Enter wildcards with \xxx (xxx in the decimal system)
- Enter \ as \\
OK
Cancel
Help

Figure 1-39 *Properties - SINAUT Dedicated Line TIM* dialog, *AT Initialization*

- *Initialization string* input box:
  In the left input box, the AT string is displayed as text and in the right-hand box in hexadecimal notation. No entries can be made in the right-hand box. In the left-hand box, a string can only be entered when either no entry can be found for the current combination of modem and network parameters in the SINAUT modem database or the *User defined* option was set. Otherwise the valid string is taken from the database.
- *User defined* option:
  It allows the manual entry of AT Initialization strings for the basic settings of the modem.
- *Type of string* output box:
  This box displays the content of the current string.

Special features of the MD4 modem

On the MD4 modem, the standard string is *ATS45=83$P1\N0&W$M=n*.
The part string *$P1* defines the V.110 mode at a transmission rate of 9600 bauds.
The *n* character at the end of the string is the placeholder for the MSN (last digit of the telephone number) of the subscriber.

The stored standard string allows only transmission rates of 9600 and 19200 bauds, no other rates are not supported. This standard setting does, however, allow communication with the following subscribers:

- Old ISDN modems installed with SINAUT ST1 devices
- GSM stations
- SMS centers

When using MD4 modems in ISDN networks, it is only possible to operate at a transmission speed of 9600 bauds with the standard setting and not at the maximum speed of 38000 bauds.

If the maximum speed of 38000 bauds is to be used with MD4 modems in pure ISDN networks, the *$P1* entry in the standard string must be replaced by *$P5*.
This changes from the V.110 transmission mode to the X.75 mode allowing a transmission rate up to 38000 bauds.

When using MD4 modems, note the following if you change the telephone number later:

Note

Since the MSN number is automatically added to the generated AT string with the MD4 modem, if you change the telephone number later, remember that the MSN number in the AT string for the MD4 modem may need to be modified manually.

1.6.8 Plausibility check of the network configuration

A plausibility check of the configured network is performed either when the network is stored with the *Save and Compile...* function or when the consistency check is started directly from the *Network / Check Consistency* menu. The following is reported:

- Subscribers not connected to a any subnet
- Subnets with only one subscriber
- Inconsistent connections, for example due to the wrong WAN protocol

The following are also checked for SINAUT networks:

- Compatibility of the connected modem types with each other
- Compatibility of the connected modem types with the network parameters

The following is checked for SINAUT dedicated line networks:

- The parameter assignment of a master station for the dedicated line network
- The existence of more than one master station for the dedicated line network in polling mode

Note

If SIMATIC S7-300 stations are connected only over the WAN, the following warning may be displayed for the non-connected MPI nodes of a CPU in older STEP 7 versions: "CPU... (Station ..): The subscriber (...) is not connected to a network."

This warning can be ignored.

To complete network configuration, the configured version must be saved with the *Network / Save* menu to allow other STEP 7 and SINAUT applications access to the configured data.

Configuration continues by calling the SINAUT configuration tool.

Note

From the network configuration, not only the *Save* but also the *Save and Compile...* function can be called that generates the system data blocks (SDB) after saving the configuration.

To acquire all the configured parameters of the TD7 software when generating the SDBs both in new projects or when making modifications to the configuration of existing SINAUT installations, the generation of SDBs for SINAUT networks should only be performed in the *Subscriber Administration* of the SINAUT configuration tool.

1.7 The SINAUT Configuration Tool

1.7.1 Working with the SINAUT ST7 configuration tool

The SINAUT ST 7 configuration software fits into the familiar Windows sequences. This means that functions such as

- Window technology
- Menu bar
- Toolbar
- Online help for the dialogs of the configuration software
- Online help for the TD7 blocks
- Printing

are integrated according to the Windows and STEP 7 standards. Working with the configuration tool is explained in the individual sections of the chapter.

General information on working with the tool

You select a menu or a graphic object by clicking once with the left mouse button.

Further functions for this object are then available over the menu bar, over the buttons of the toolbar or often over a context menu that opens when you select an object with the right mouse button.

The buttons for the properties dialogs available for configuration have the following functions:

- *OK:* Confirms the entries made and closes the dialog.
- *Cancel:* Entries made are ignored and the dialog is closed.
  The *Cancel* button is not available in the dialogs for subscriber administration in the SINAUT configuration tool. In this case, the dialog is closed without entering changes by clicking on the close button [x] in the right-hand top corner of the header line of the dialog.
- *Help:* Opens the online help.

1.7.2 Starting the SINAUT Configuration Tool

Introduction

The SINAUT ST 7 Configuration Tool is started in the *Start / SIMATIC / SINAUT ST7 / Configuration* menu.

At the start of configuration, you open a project with the *Project / Recently Used* menu or the *Open Project* button in the toolbar.

After selecting the project, the *SINAUT ST7: Configuration* dialog opens making the three following main functions of the SINAUT Configuration Tool available:

- *Connection Configuration* for SINAUT connections
- *Subscriber Administration* for SINAUT subscribers
- *SINAUT ST1 - Configuration Overview*

  For more information on SINAUT ST1, refer to release 05/2007 of the manual.

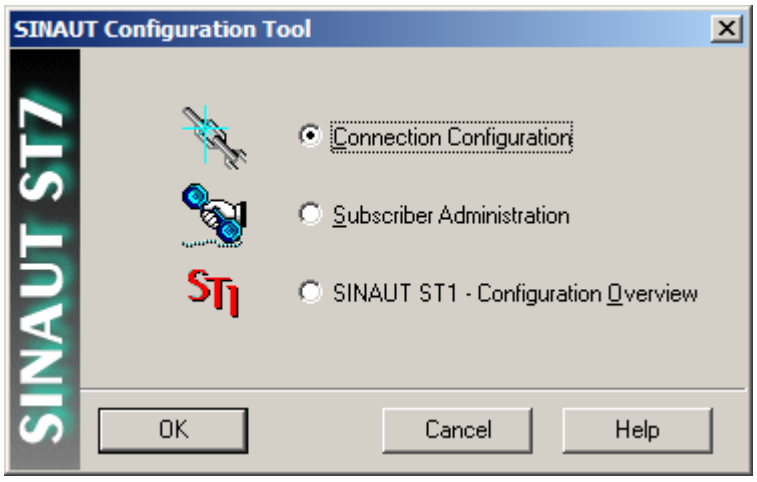


Figure 1-40 Selection dialog of the SINAUT Configuration Tool

To continue configuration, first select the *Connection Configuration* option.

When the SINAUT Configuration Tool is open, you can change between the three functions using:

- The *SINAUT / ...* menu,
- The corresponding buttons in the toolbar or
- the following function keys:
  - F3 for *Connection Configuration*
  - F4 for *Subscriber Administration*

Program information and SINAUT Internet pages

You can display program version and copyright information in the About dialog of the SINAUT configuration tool that can be opened using the *Help / About* menu.

With the Internet address *www.sinautst.de* at the top right in the SINAUT About dialog, you can open the SINAUT home page directly.

1.8 Configuring connections in the SINAUT Configuration Tool

1.8.1 Configuring SINAUT connections

The SINAUT ST7 connections

In SINAUT ST7 installations, connections are always configured between the following subscribers:

- From CPU to CPU
- From CPU to a SINAUT ST7cc/ST7sc control center

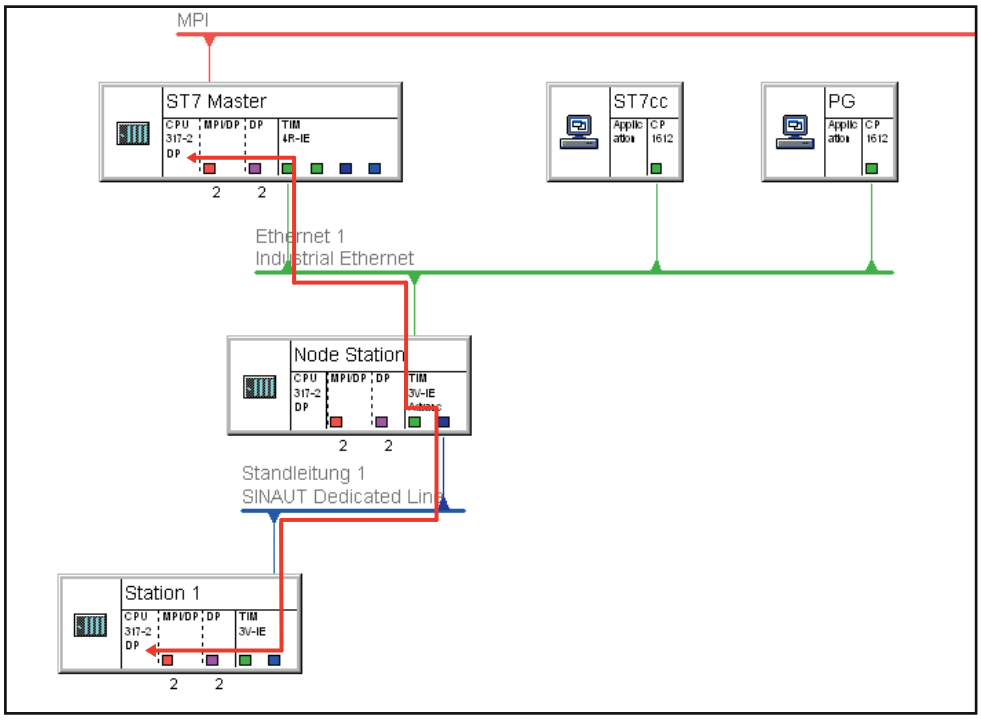


Figure 1-41 Example of a SINAUT ST7 connection from the master station to station 1

The connection shown in dark red from the master station to station 1 is not visualized in this form in the network configuration with NetPro.

This is taken into account in the connection configuration that allows the user

- to configure connections without knowing the exact connection path and
- to view the exact connection path over a list of used network nodes. The degree of detail of the connection information can be set by users themselves.

View of the *Connection Configuration* window

After opening the connection configuration, the configuration window opens.

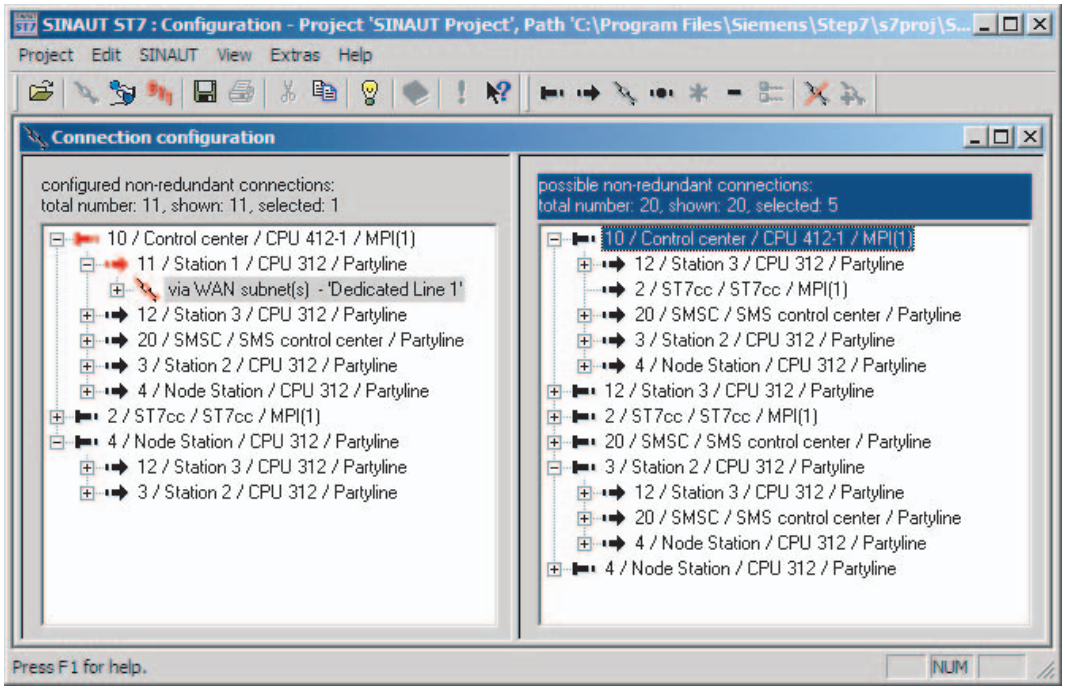

SINAUT ST7 : Configuration - Project 'SINAUT Project', Path 'C:\Program Files\Siemens\Step7\s7proj\S...
Project Edit SINAUT View Extras Help
Connection configuration
configured non-redundant connections:
total number: 11, shown: 11, selected: 1
10 / Control center / CPU 412-1 / MPI(1)
11 / Station 1 / CPU 312 / Partyline
via WAN subnet(s) - 'Dedicated Line 1'
12 / Station 3 / CPU 312 / Partyline
20 / SMSC / SMS control center / Partyline
3 / Station 2 / CPU 312 / Partyline
4 / Node Station / CPU 312 / Partyline
2 / ST7cc / ST7cc / MPI(1)
4 / Node Station / CPU 312 / Partyline
12 / Station 3 / CPU 312 / Partyline
3 / Station 2 / CPU 312 / Partyline
possible non-redundant connections:
total number: 20, shown: 20, selected: 5
10 / Control center / CPU 412-1 / MPI(1)
12 / Station 3 / CPU 312 / Partyline
2 / ST7cc / ST7cc / MPI(1)
20 / SMSC / SMS control center / Partyline
3 / Station 2 / CPU 312 / Partyline
4 / Node Station / CPU 312 / Partyline
12 / Station 3 / CPU 312 / Partyline
2 / ST7cc / ST7cc / MPI(1)
20 / SMSC / SMS control center / Partyline
3 / Station 2 / CPU 312 / Partyline
12 / Station 3 / CPU 312 / Partyline
20 / SMSC / SMS control center / Partyline
4 / Node Station / CPU 312 / Partyline
4 / Node Station / CPU 312 / Partyline
Press F1 for help.
NUM

Figure 1-42 The *Connection Configuration* window of the SINAUT Configuration Tool

In the right-half of the window, the *possible connections* are listed in a tree structure resulting from the network configuration in NetPro. The connections actually required and used for communication in the SINAUT installation must be transferred from the right-hand to the left-hand window *Configured connections*.

The two lists display the number of connections involved.

The entries in the connection tables must be interpreted as follows:

Table 1- 2 Symbols in the connection list of the connection configuration

Level	Symbol	Meaning
1		Connection starting point
2		Connection end point
3		Alternative path
4		Connection node over which the connection runs

Invalid connections are displayed in red as shown in the example of a connection that no longer exists due to reconfiguration.

The labeling of the individual connection point in the basic setting describes the relevant subscriber with:
Subscriber number / Station name / Module / Interface.
Example: *5 / Station 3 / CPU 312 / MPI (2)*

The representation can be set to meet individual requirements using the *Extras / Options* menu.

Functions of connection configuration

To make configuration of the required connections as simple as possible for the user, the SINAUT Configuration Tool uses the following strategy:

- The entire currently configured network is analyzed. All potential communication subscribers from the SINAUT perspective are assigned a subscriber number if they do not already have one. The subscriber numbers for CPU modules and third-party stations are assigned starting at no. 1, for TIM modules there are assigned starting at no. 1001.
- A tracking algorithm detects ALL connections in the current network. These connections can also extend over several LANs and SINAUT networks. The connections permitted based on specified rules are represented on the right as *possible connections* in a tree structure.
- SINAUT connections that have already been configured are displayed in the left-hand window for *configured connections*. Each of the connections loaded there is then checked to establish whether its configured parameters match the current network and hardware configuration. If this is not the case, and error message indicating incorrect connections is displayed as soon as the connection configuration is opened and the bad connection is displayed in red in the *configured connections* window.

If a station of the type *other station* or *SIMATIC S5* was configured in NetPro, connections from the stations to stations of the type ST7, PG/PC, or an ST7cc control center are not displayed. This does not, however, mean that these connections do not exist. These connections are in fact displayed in the opposite direction; this means, for example, from an ST7 station to a station of the type *other station* or *SIMATIC S5*.

As a general rule, a connection displayed in the *configured connections* in only one direction, works in both directions.

Selecting the required CPU-CPU connections

If no connections are displayed in the left-hand window, the required connections must be transferred from the right-hand window. Follow the steps outlined below:

1. Expand the tree structure by clicking on the branch symbol (+) or by double-clicking on the connection group. The tree structure opens.
2. Select a *possible connection* in the right-hand window.
3. Enter the possible connection as a *configured connection* in the left-hand window by
   - selecting the *Edit / Apply* menu or
   - pressing the right mouse button and selecting *Apply* in the displayed context menu.

If alternative communication paths exist and you want to use them, expand the possible connection structure in the *possible connections* by double-clicking on it and select the connection and apply it.

Redundant connections:
With redundant connections, for example those of a redundant ST7cc/ST7sc control center, the upper connection in the tree structure is the preferred path and the lower connection is the substitute path. The connection with the preferred path should therefore be applied first.

Connections that are not required can be removed from the list of configured connections at any time. To do this, select the connection in the *configured connections* and
select the *Edit / Delete* menu or
select *Delete* in the displayed context menu (right mouse button)

Bad connections displayed in the *configured connections* window are shown in red and can result from bad configuration or subsequent changes to a configuration. If there is an incorrect connection between two nodes, the old and no longer valid connection must be deleted from the *configured connections* and the current connection must be taken from the list of *possible connections* again.

Note

If a connection configured in the network configuration is not included in the list of *possible connections* because it is invalid, this is displayed in the list of *invalid connections*. The *invalid connections* dialog is described separately and contains information on connections that do not conform with SINAUT.

Saving the connection configuration

Once any invalid connections have been checked and removed and all required connections configured so that they appear in the left-hand window, the connection configuration must be saved with the *SINAUT / Save* menu or the *Save* button in the toolbar. Saving is necessary to store the connections permanently.

If a message is displayed during saving indicating that a connection between two stations configured in NetPro could not be found, the connection must be checked in NetPro and reconfigured.

After saving the connection configuration, open *Subscriber Administration* of the SINAUT Configuration Tool to configure the subscriber data and to generate the system data blocks there.

Changing the connection configuration

By opening the connection configuration again, you can change the scope of the configured connections at any time. By changing parameter settings or by reconfiguring, it is possible that a previously configured connection no longer exists. This then appears in the *Recover lost connections* list that is described separately.

After changing the connection configuration, this must be saved, *Subscriber Administration* must be called and the generation of the system data blocks started.

1.8.2 Invalid Connections

With its algorithms, connection configuration finds all the possible connections in the current project. Connections that do not meet certain rules are displayed for the user in the *Invalid*

Connections dialog. The connections contained here are then not included in the list of *possible connections*.

The *Invalid Connections* dialog is displayed using the *SINAUT / Show Invalid Connections* menu or the *Show Invalid Connections* button in the toolbar.

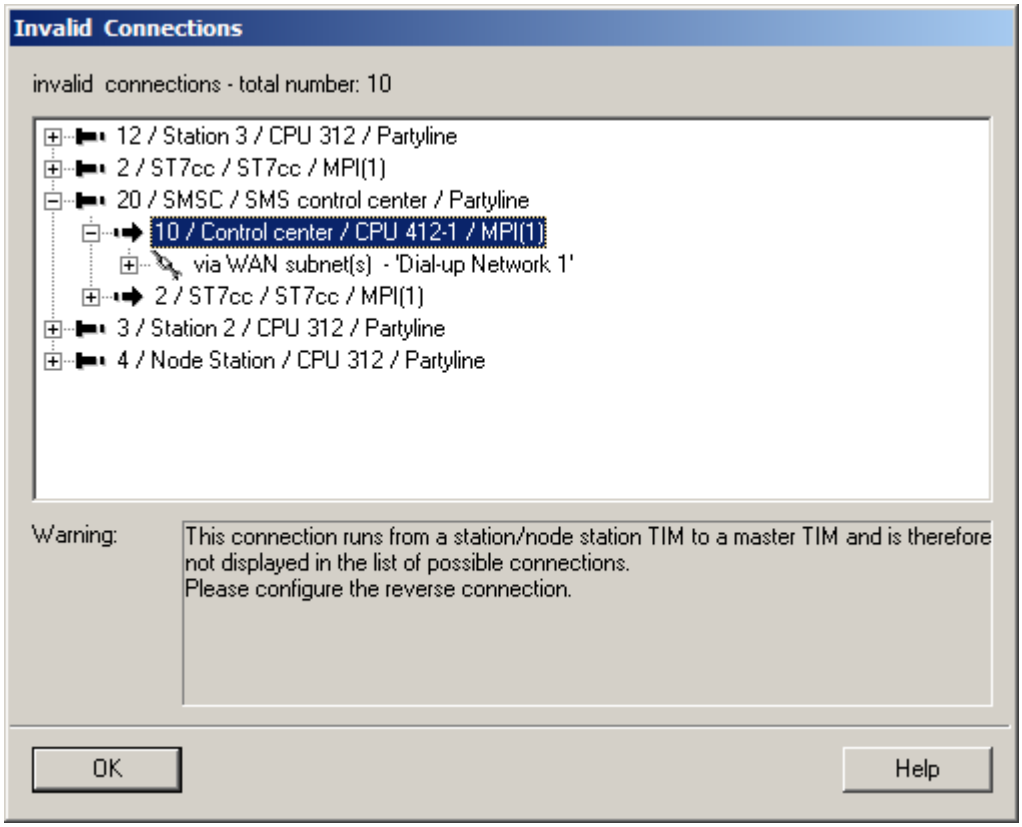

Invalid Connections
invalid connections - total number: 10
12 / Station 3 / CPU 312 / Partyline
2 / ST7cc / ST7cc / MPI(1)
20 / SMSC / SMS control center / Partyline
10 / Control center / CPU 412-1 / MPI(1)
via WAN subnet(s) - 'Dial-up Network 1'
2 / ST7cc / ST7cc / MPI(1)
3 / Station 2 / CPU 312 / Partyline
4 / Node Station / CPU 312 / Partyline
Warning:
This connection runs from a station/node station TIM to a master TIM and is therefore not displayed in the list of possible connections.
Please configure the reverse connection.
OK
Help

Figure 1-43 *Invalid Connections* dialog

If you expand a connection structure in the list of *invalid connections* by double-clicking on it and if you then select a single connection, the reason for the invalidity and a note on how to remedy the situation are displayed in the lower part of the dialog. The note might, for example, inform you that the connection should be configured in the reverse direction.

Within SINAUT ST7, permitted connections must adhere to the following rules:

- A connection must not run through an inconsistent network. Examples are described along with the plausibility check in the network configuration.
- A connection must not run through a WAN sub-connection on which two MD3 modems communication each other using 1200 bauds / half duplex / AT mode.
- A connection should not run from a station or node station TIM to a master TIM. The reverse direction is preferable.

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

Note

To avoid including large numbers of connections unnecessarily in the list of possible and configured connections, some connections used between two subscribers in both directions are shown only in one direction.

A connection displayed in the configuration tool is always valid in both directions.

1.8.3 Recovering lost connections

If changes are made to connections in a project, it is advisable to open the *Recover lost connections* window afterwards. You can do this with:

- The *SINAUT / Recover lost connections* menu or
- The *Recover lost connections* button on the right of the toolbar

If changes to the connection configuration, previously existing connections were modified and either linked to other objects or completely deleted, connection configuration has algorithms with which to find these lost connections in the project.

Deleted connections are displayed in red in the left-hand part of the *Recover lost connections* window. Connections that are similar to the deleted connections may also be listed in the right-hand part of the window. This allows you to check whether there is a substitute or successor to the last connection.

If you no longer require the lost connections in the left-hand part of the window, you can delete these by selecting and clicking the *Delete lost connections* button.

If connections are displayed in the right-hand part of the window that are similar to the last connections you can insert these again if you have accidentally deleted connections and still require them. To do this, select the connection in the right-hand part of the window and click the *Recover connections* button.

1.8.4 Printing connection lists

To document the configured ST7 connections, the SINAUT configuration tool allows connection lists to be printed in two formats. You start a printout with the *Project / Print* menu.

Before printing, you can use a print preview function to check the printout using the *Project / Print Preview* menu.

1.9 Subscriber administration in the SINAUT configuration tool

Once the SINAUT ST7 connections have been configured, all the requirements are met so that you can

- create,
- display,
- process and
- pack the subscriber data for the ST 7 communication subscribers so that it can be understood by the hardware components; in other words, in data blocks (DBs) or system data blocks (SDBs).

Processing is always necessary when data needs to be acquired that is connection-related; in other words cannot be assigned to a particular subscriber.
Processing of subscriber data is also necessary when SINAUT data is stored for non-SINAUT objects, for example the DB configuration for the CPU modules and SMS messages.
How to set parameters for this data in the subscriber administration of the SINAUT configuration tool is described below.

After starting subscriber administration, first the previously known data of the subscribers is loaded from the data management and then updated. The following data is updated:

- Subscriber information: Which subscribers exist?
- Networking information: Who communicates with whom over which connections?
- DB configuration information: Which data blocks are generated for a CPU?

The subscriber list always shows the latest situation in the SINAUT subscriber world.

1.9.1 Subscriber list

Information in the subscriber list

In the left-hand window, the subscriber administration contains a tree structure of the subscriber types of the project and in the right-hand window the subscriber list itself.

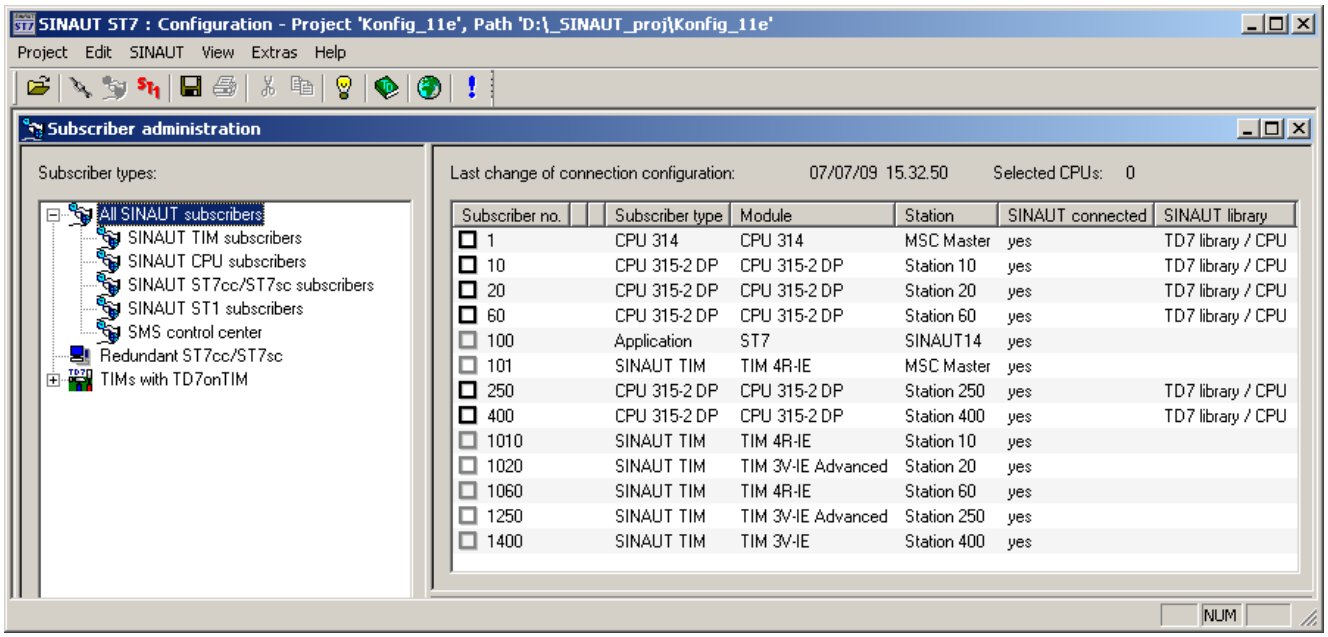

SINAUT ST7 : Configuration - Project 'Konfig_11e', Path 'D:_SINAUT_proj\Konfig_11e'
Project Edit SINAUT View Extras Help
Subscriber administration
Subscriber types:
All SINAUT subscribers
SINAUT TIM subscribers
SINAUT CPU subscribers
SINAUT ST7cc/ST7sc subscribers
SINAUT ST1 subscribers
SMS control center
Redundant ST7cc/ST7sc
TIMs with TD7onTIM
Last change of connection configuration: 07/07/09 15.32.50 Selected CPUs: 0
Subscriber no. | Subscriber type | Module | Station | SINAUT connected | SINAUT library
1 | CPU 314 | CPU 314 | MSC Master | yes | TD7 library / CPU
10 | CPU 315-2 DP | CPU 315-2 DP | Station 10 | yes | TD7 library / CPU
20 | CPU 315-2 DP | CPU 315-2 DP | Station 20 | yes | TD7 library / CPU
60 | CPU 315-2 DP | CPU 315-2 DP | Station 60 | yes | TD7 library / CPU
100 | Application | ST7 | SINAUT14 | yes
101 | SINAUT TIM | TIM 4R-IE | MSC Master | yes
250 | CPU 315-2 DP | CPU 315-2 DP | Station 250 | yes | TD7 library / CPU
400 | CPU 315-2 DP | CPU 315-2 DP | Station 400 | yes | TD7 library / CPU
1010 | SINAUT TIM | TIM 4R-IE | Station 10 | yes
1020 | SINAUT TIM | TIM 3V-IE Advanced | Station 20 | yes
1060 | SINAUT TIM | TIM 4R-IE | Station 60 | yes
1250 | SINAUT TIM | TIM 3V-IE Advanced | Station 250 | yes
1400 | SINAUT TIM | TIM 3V-IE | Station 400 | yes
NUM

Figure 1-44 Windows of subscriber administration

By making a selection in the directory tree in the left-hand window, only certain subscriber types can be displayed in the subscriber list. The *TD7 on TIM* folder in the directory tree is used to configure the TD7 software for an Ethernet TIM and is not relevant for the subscriber parameter assignment described here.

In the subscriber list on the right, you will see the following entries for the SINAUT ST7 communication subscribers:

- **Subscriber no.**: The subscriber number of the SINAUT subscriber that is unique throughout the project and is required for WAN communication. This has an extra check box to allow the CPU modules to be selected for system data generation.
- **Red. Subscriber no.**: The *Redundant subscriber number* parameter is used only when there is a redundant partner for the subscriber in question. The number specifies the common subscriber number under which the redundant system can be addressed by other subscribers.
- **Subscriber no. of red. Partner**: The *Subscriber number of the redundant partner* parameter is used only when there is a redundant partner for this subscriber. The parameter specifies which of the subscribers belong to a redundant relationship.
- **Subscriber type**: The *subscriber type* specifies the class of subscriber involved. This cannot be changed by the user.
- **Module**: The module, application or PC/PG name. This can be changed in the configuration. As default, this is the name of the module type or the application as specified in the configuration.

- **Station**: Name of the station assigned in the network configuration in *NetPro*.
- **SINAUT connected**: Specifies whether a SINAUT connection was configured for the subscriber.
- **SINAUT library**: Name of the SINAUT software block library for CPU and TD7onTIM-compliant TIM modules.

The setting of the subscriber-related properties is made in the *Properties of subscriber* dialog that is opened when you double-click on one of the subscribers in the list.

Marking for selective system data generation

In the subscriber administration, it is possible to select any CPU subscriber for later selective generation of the software blocks. To select a subscriber, click in the check box in front of the subscriber number (a check mark is set). The number of selected CPU modules is displayed in the text box *Selected CPUs:* above the list. You can remove the check mark again by clicking with the mouse.

Changing the subscriber number

To change the subscriber number, click on the required subscriber. By clicking again on the subscriber number, by typing *Alt+Return*, by pressing function key F2 or by selecting the menu *Edit / Change Subscriber*, the *Subscriber number* field becomes editable. The user can then enter any unassigned subscriber number as required.

Dialogs below the subscriber list

Any dialogs displayed below the subscriber list are used for configuring the TD7 software for the TIM (TD7onTIM) that can only be configured for TD7onTIM-compliant TIM modules. These dialogs are not available for other subscribers. The TD7onTIM is configured and the parameter settings made following this in the subscriber parameter assignment.

Generating a redundant ST7cc/sc control center

If two SIMATIC PC stations were configured for a redundant SINAUT ST7cc/sc control center, the assignment of the two redundant partners is made at this point. To do this, select the entry "Redundant ST7cc/ST7sc server" in the directory tree and select "Add redundant ST7cc/ST7sc..." from the shortcut menu (right mouse button).

You can delete a redundancy grouping again by selecting the entry of the redundancy grouping in the directory tree and then selecting the "Delete redundant object" shortcut menu command. This deletes only the data of the grouping but not the underlying subscribers.

For detailed information on configuring a single or redundant ST7cc/ST7sc control center, refer to the SINAUT ST7cc or ST7sc documentation.

1.9.2 Configuring the GPRS providers for MSC stations

Configuring GPRS providers

If you want to use the GPRS service with the MSC protocol, you will need to configure at least one GPRS provider for the MSC subscribers. To avoid having to call up the dialogs for configuring the MSC subscribers more than once in the SINAUT configuration tool, we recommend that you configure the providers first. Click on the globe icon ( ).

The "List of GPRS Providers" dialog opens.

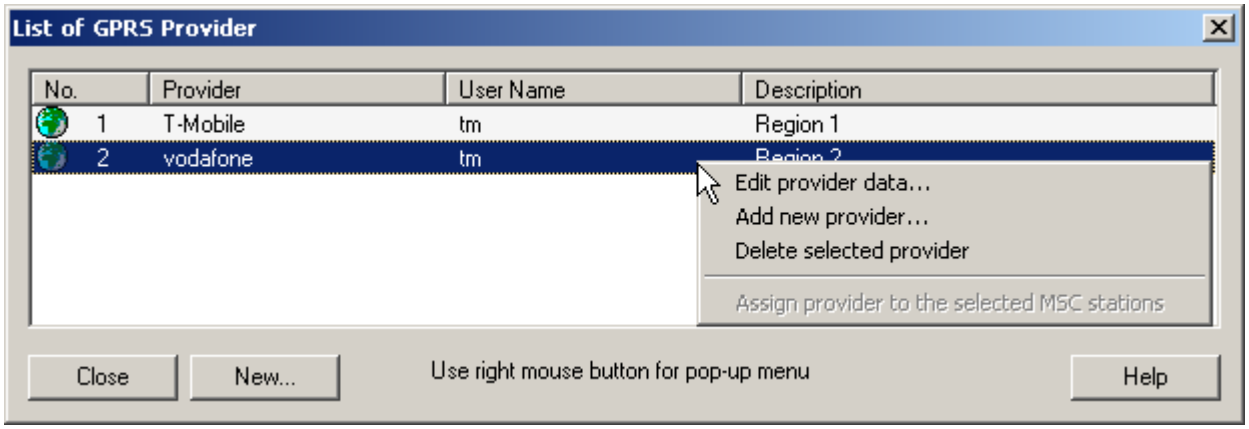


Figure 1-45 "List of GPRS Providers" dialog box

Creating GPRS providers in the "List of GPRS Providers" dialog

This dialog displays the configured GPRS providers with their names, the user name and a description.

Apart from the "New..." button for creating a new provider (see below), you can also use the following four shortcut menu commands in the dialog:

- "Edit provider data..."

  The "Provider Properties" dialog opens. Here, you can configure the properties required for the GPRS provider (see below: The "Provider Properties" dialog).

- "Add new provider..."

  The "Provider Properties" dialog opens. Here, you can configure a new GPRS provider with the required properties (see below: The "GPRS Provider Properties" dialog).

- "Delete selected provider"

  The provider you have selected is deleted from the list.

- "Assign provider to the selected MSC stations"

  See below for a description.

Meaning of the globe icon

The globe icon in this dialog has a the following meaning depending on its color:

- Green ( )

  All the required properties of the provider are configured.

- Black and white ( )

  The configured data of the provider is not complete.

The "GPRS Provider Properties" dialog

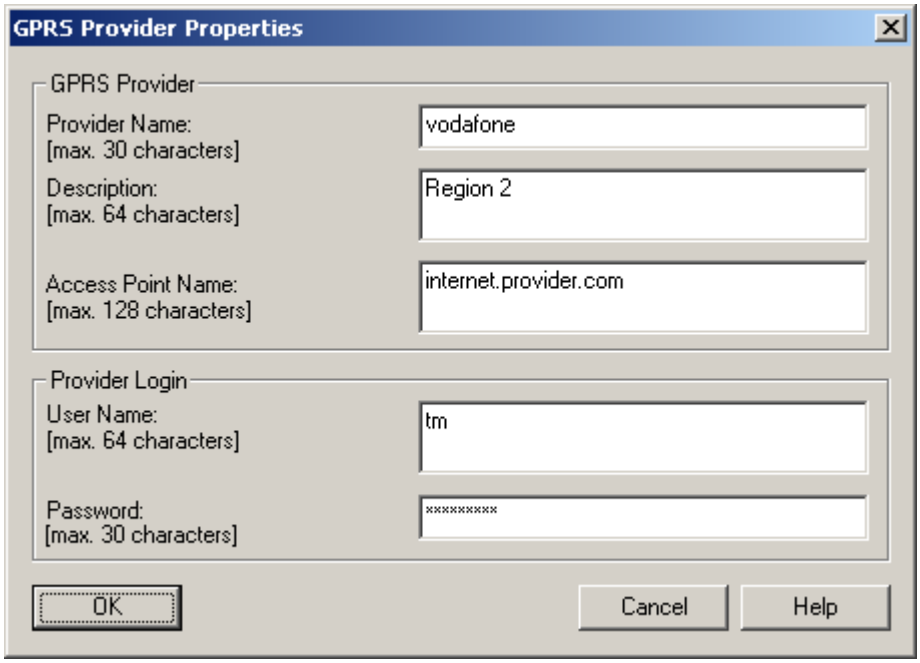


Figure 1-46 The "GPRS Provider Properties" dialog

Configure the following properties for the GPRS provider:

- Provider name

  Here, you need to assign a name for the provider. You can choose any name.

- Description

  Here, you can enter a description of the provider if required. This could, for example, be a assignment to the stations being served.

- Name of the Internet access

  Here, enter the name of the Internet access (APN) of your provider in the GPRS network. The provider will give you this name.

- User name *)

  Here, enter the user name with which the connection to the provider will be established. The provider will give you this name.

- Password *)

  Here, enter the password required to log on with the provider. If the provider does not demand a password, you could, for example, enter the word "guest" here.

*) There must be no blanks in the user name or password.

"Delete selected provider"

If you select a provider in the "List of GPRS Providers" dialog, and then select the "Delete selected provider" shortcut menu command, the provider will be deleted from the list.

Deleting is only fully complete after you exit the dialog with "OK".

"Assign provider to the selected MSC stations"

You can only assign a GPRS provider to one or more stations when the properties dialog of this station is open in the SINAUT configuration tool at the "MSC Station List" and you have selected the relevant menu command there.

The provider assignment is described in the following section:

"MSC Station List" (Page 110)

1.9.3 The "Properties of Subscriber" dialog - parameter overview

Overview of the tabs of the *Properties of subscriber* dialog

By double-clicking on a subscriber number or by selecting the *Edit / Properties* menu, the *Properties of subscriber* dialog is displayed. The content and number of the tabs of the *Properties of subscriber* dialog depend on the subscriber type.

Table 1- 3 Overview of the Properties dialog tabs according to subscriber type

Tab	CPU	TIM	Other station, SIMATIC S5, PC station, PG	Names of follow-up dialogs
Info	yes	yes	yes	-
Partner	yes	yes	-	-
Connections	yes	yes	-	Properties - Local Connection
Polling List	-	Master TIM	-	Properties - Poll list entry
MSC station list	-	Master TIM (MSC master)	-	Properties - MSC master Properties - MSC station Provider assignment
Telephone Directory	-	When necessary	-	Properties - Telephone number

Tab	CPU	TIM	Other station, SIMATIC S5, PC station, PG	Names of follow-up dialogs
DB Configuration	yes	-	-	-
SMS Configuration	When SMS center configured	-	-	SMS CPU Configuration, SMS DB Data, SMS Message Data
Library Information	yes	-	-	-

1.9.4 "Info" tab

Info tab

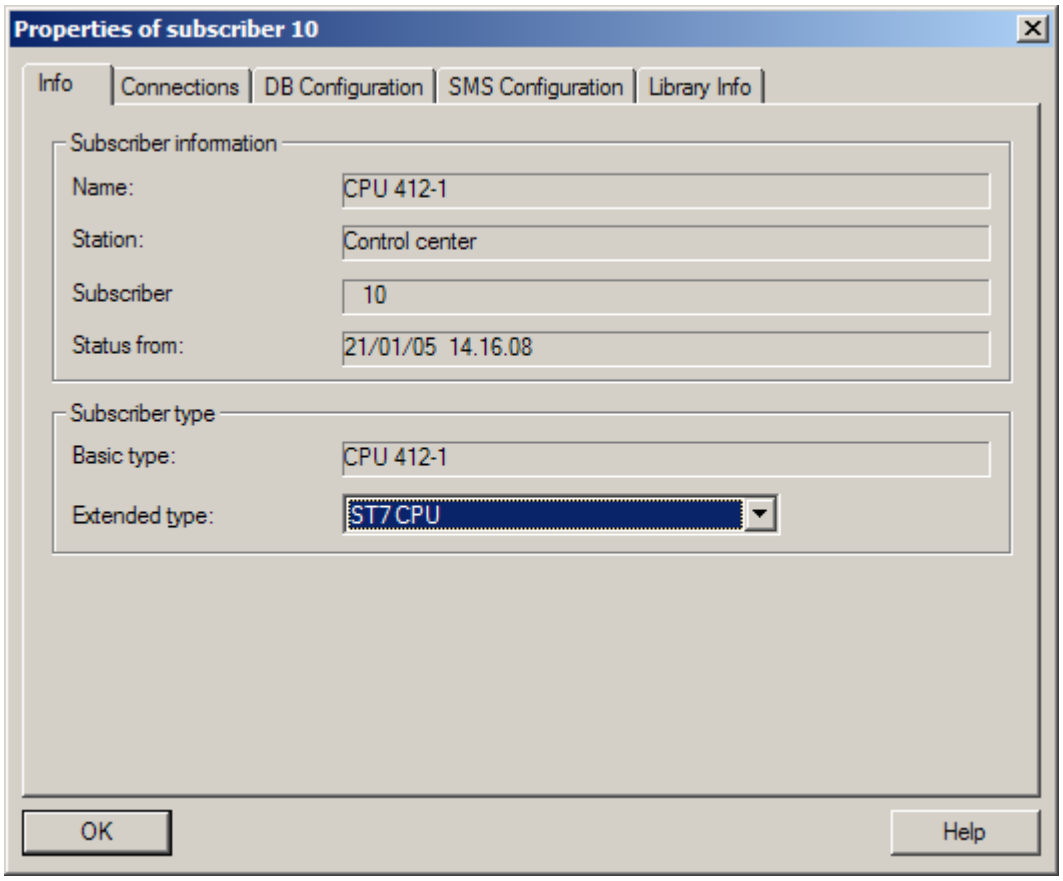


Figure 1-47 *Properties of subscriber* dialog (CPU), *Info* tab

The *Info* tab displays the following information on the selected subscriber:

- *Name* shows the default name of the module or the name assigned in network configuration
- *Station* displays the set network node type
- The *subscriber number* is displayed
- *Status from* displays the date of the last configuration
- The *date created* is displayed.
- *Basic type* displays the network object type from the network configuration
- *Extended type* displays the extended network object type adopted in the network configuration (for example ST7-CPU, ST7-TIM, ST7cc or SMS center)

1.9.5 "Partners" tab

Partners tab

This tab lists the connection partners of the selected module with subscriber number, module type, name and station name.

1.9.6 "Connections" tab

Connections tab

This tab lists all the configured local connections over LAN along with their most important properties for the current subscriber:

- X connections

  Unconfigured S7 connections that use the SFCs "X_SEND" and "X_RCV".
- PBC connections

  Configured S7 connections that use the SFBs "BSEND" and "BRCV".
- CR connections

  Read/write connections of the TD7 software "TD7onTIM" to the local CPU that do not require S7 connections.
- MSC connections

  Connections for the MSC protocol that do not require S7 connections.

Note

During the analysis of the subscriber data, if the configuration software detects that STEP 7 homogeneous connections are necessary for processing a SINAUT ST7 connection, these are created automatically. These are connections from the S7-400 CPU to a TIM module and from TIM to TIM module over the MPI bus and over communication block connections.
As an alternative, the user can create these connections manually during network configuration. Existing connections are automatically used by the SINAUT configuration tool.

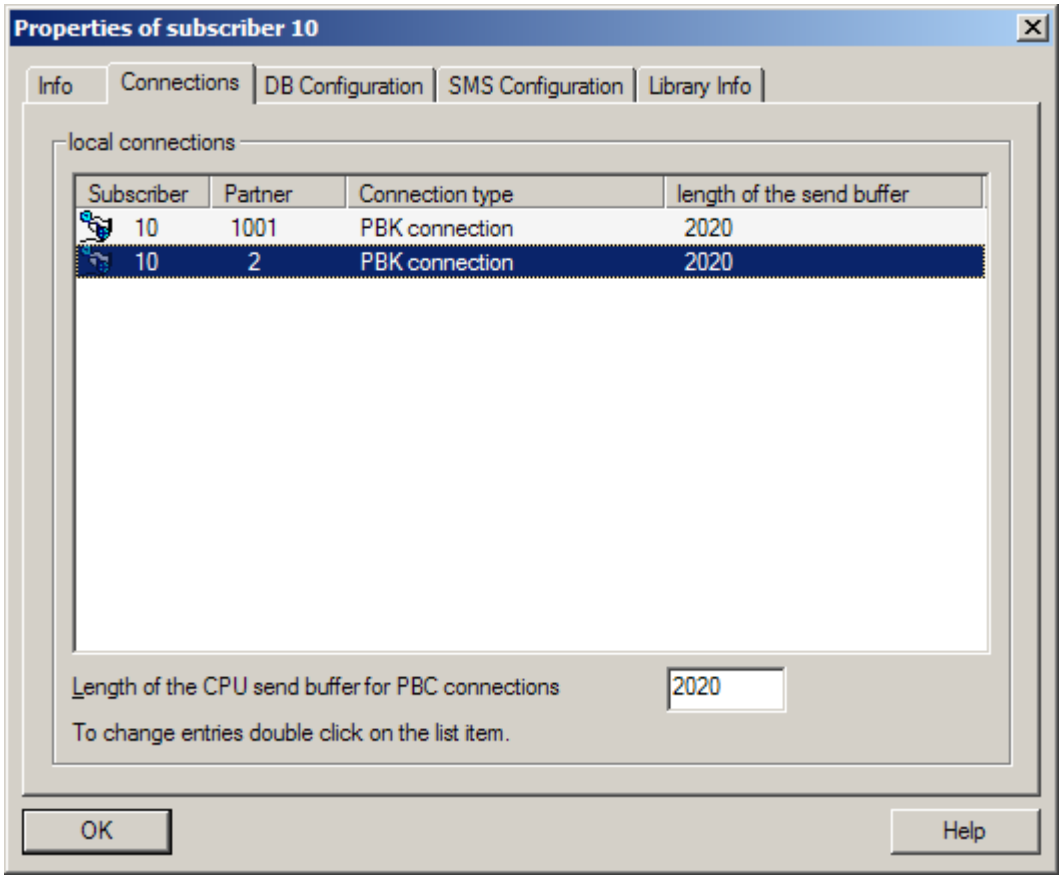


Figure 1-48 *Properties of subscriber* dialog (CPU), *Connections* tab

Here, you can configure the following:

- The length of the CPU send buffer for any existing communication function block connections of a CPU. This is the same for all communication function block connections of the current CPU.
  Range of values: 202 .. 65208 bytes
  Default: 2020 bytes

By double-clicking on a subscriber row in the *local connections* output box, you open the *Properties - Local Connection* dialog for this connection.

Properties - Local Connection dialog for the TIM

This dialog displays the properties of the local LAN connection of a TIM module to its CPU.

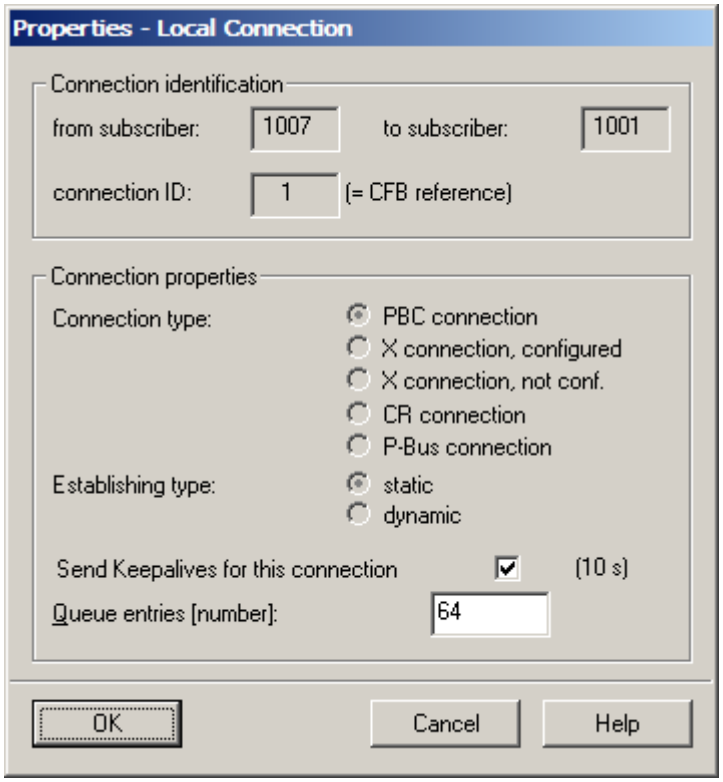


Figure 1-49 *Properties - Local Connection* dialog (TIM)

For the local connection of the TIM module, you can set the following here:

- *Send Keepalives for this connection:*
  If this option is enabled, keepalive messages are sent on this connection at the intervals set in the network configuration for the TIM.
- *Queue entries [number]*:
  Range of values: 10 .. 256
  Default: 64

  The number of queue entries is the number of messages that the TIM module can buffer prior to transmission. In general, you do not need to change the default setting. It may be useful to increase the value for a master TIM to relieve message traffic if there is a heavy load at certain times due to the transfer of large amounts of data, for example archive data.

Properties - Local Connection dialog for the CPU

This dialog visualizes the properties of a local LAN connection of a CPU and allows the following properties to the selected:

- CPU modules with X connections / P bus connections:
  - The *length of the send buffer [bytes]* for these connections
    Range of values: 76 ... 65382 bytes

Default: 760 bytes
This parameter is not relevant for PBC connections.

– The *Connection monitoring time [s]*, in other words, the time that must elapse before dummy messages are sent to check the connection.
Range of values: 1 ... 32 s
Default: 5 s

- CPU modules with PBC connections:
  – The *Connection monitoring time [s],*
  Range of values: 1 ... 32 s
  Default: 5 s

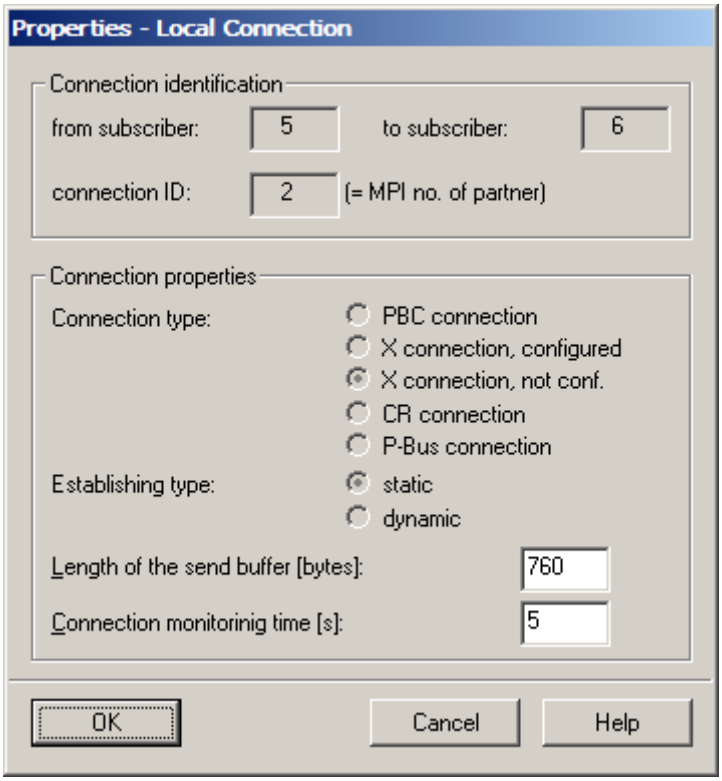


Figure 1-50 *Properties - Local Connection* dialog (CPU)

1.9.7 "Polling List" tab

Polling List tab

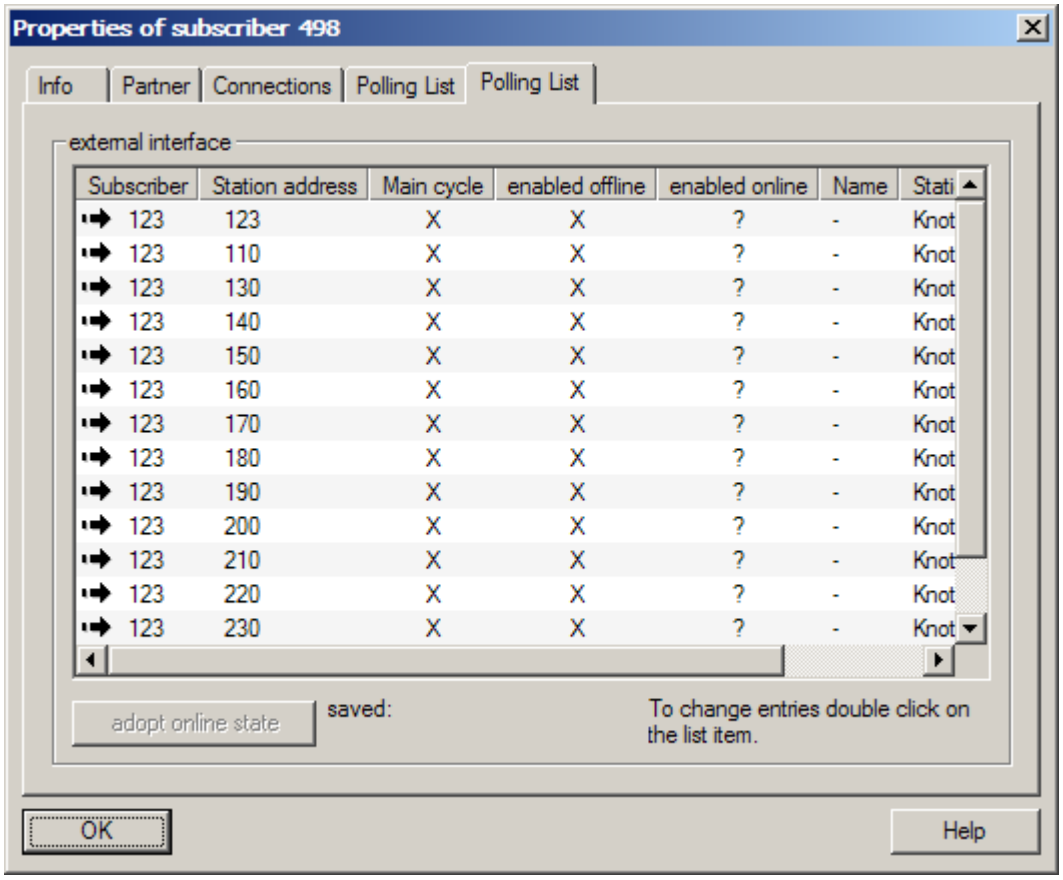

Properties of subscriber 498
Info
Partner
Connections
Polling List
Polling List
external interface
Subscriber
Station address
Main cycle
enabled offline
enabled online
Name
Stati
123 123 X X ? - Knot
123 110 X X ? - Knot
123 130 X X ? - Knot
123 140 X X ? - Knot
123 150 X X ? - Knot
123 160 X X ? - Knot
123 170 X X ? - Knot
123 180 X X ? - Knot
123 190 X X ? - Knot
123 200 X X ? - Knot
123 210 X X ? - Knot
123 220 X X ? - Knot
123 230 X X ? - Knot
adopt online state
saved:
To change entries double click on the list item.
OK
Help

Figure 1-51 *Properties of subscriber* dialog (TIM), *Polling List* tab

This tab for a TIM module displays the TIM modules that can be polled by this station in polling mode with the subscriber name, station address, ID for main cycle poll, enable status offline, enable status online, module name, and station name.

By clicking *adopt online state*, the entire online status is adopted in the configuration. This function is available only for the TIM 4R-IE.

By double-clicking on a subscriber role in the *internal interface* or *external interface* output box, the *Properties - Poll list entry* dialog for this connection opens.

Properties- Poll list entry dialog

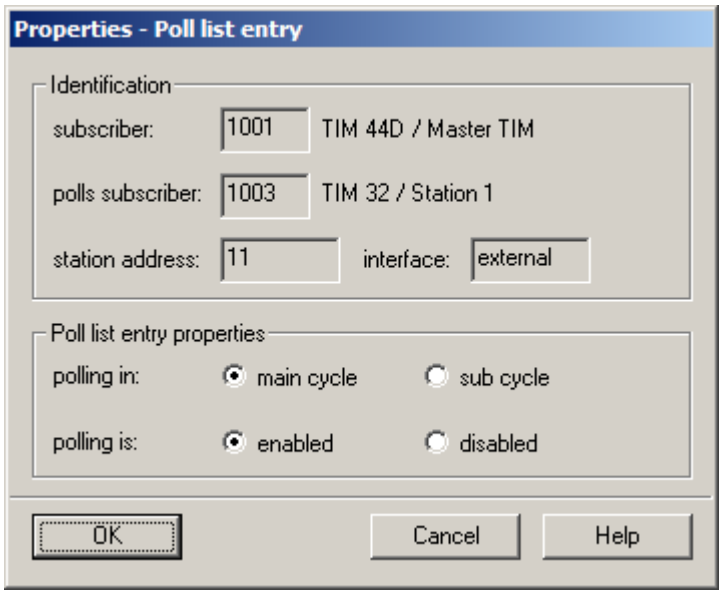


Figure 1-52 *Properties- Poll list entry* dialog

The *Identification* area displays the following properties:

- *subscriber*:
  The *subscriber* output box displays the polling subscriber (master TIM) with its subscriber number, module name, and station name.
- *polls subscriber*:
  The *polls subscriber* output box displays the polled subscriber with its subscriber number, module name, and station name.
- *station address*:
  The *station address* output box displays the station address of the polled subscriber.
- *interface*:
  The *interface* output box displays the interface type (internal/external) of the polled subscriber.

The *Properties* area displays the following options:

- *polling in*:
  Options: *main cycle*, *sub cycle*
  Depending on the selected option, the CPU module is polled in the main or sub cycle.
- *polling is*:
  Options: *enabled*, *disabled*
  If the *enabled* option is selected, polling the CPU module in polling mode is enabled. Otherwise polling is disabled.

1.9.8 "Telephone Directory" tab

Telephone Directory tab

This tab of a TIM module displays the subscribers of a dial-up network with the subscriber number, station address, dial string, enable status offline, enable status online (TIM 4R-IE only), module name and station name. If the TIM module is connected to two dial-up networks as is the case with a node station, the subscribers of both dial-up networks are listed.

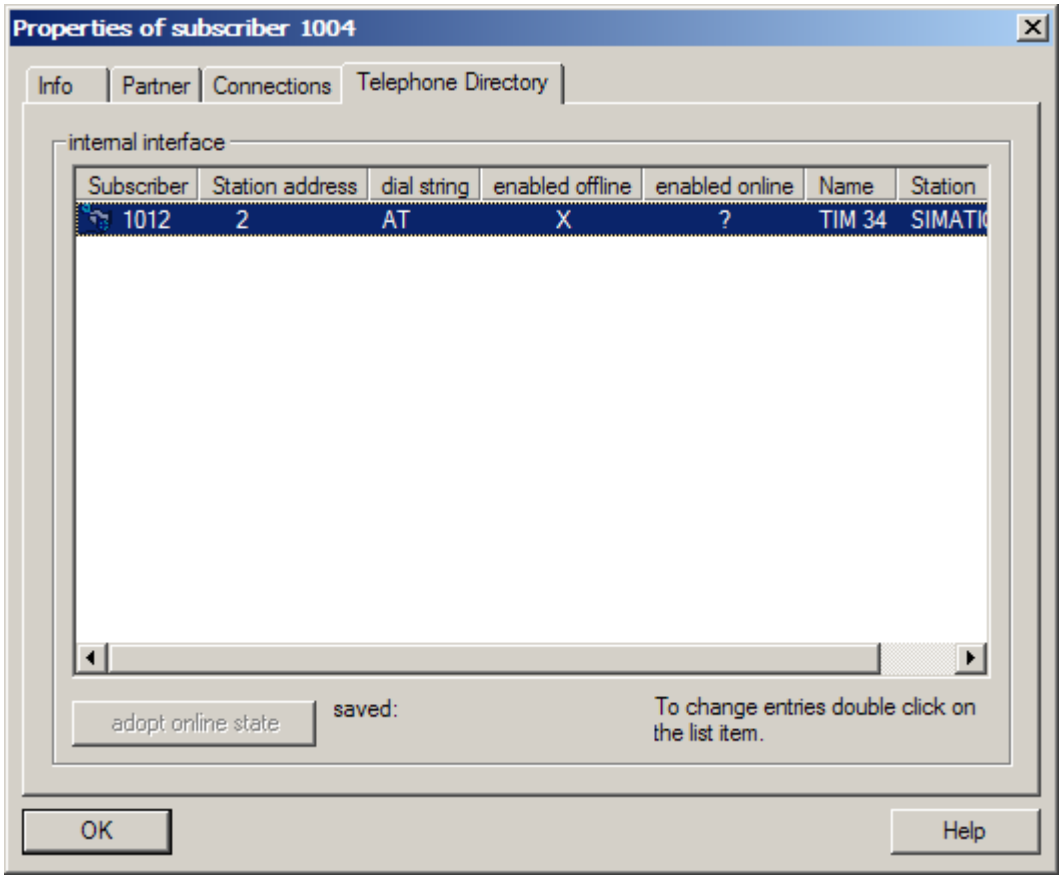


Figure 1-53 *Properties of subscriber* dialog (TIM), *Telephone Directory* tab

By clicking adopt online state, the entire online status is adopted in the configuration. This function is available only for the TIM 4R-IE.

Double-clicking on a subscriber in the *internal interface* output box opens the *Properties - Telephone Number* dialog for the connection of the subscriber selected in subscriber administration to the subscriber selected here in the *Telephone Directory* tab.

Properties- Telephone Number dialog

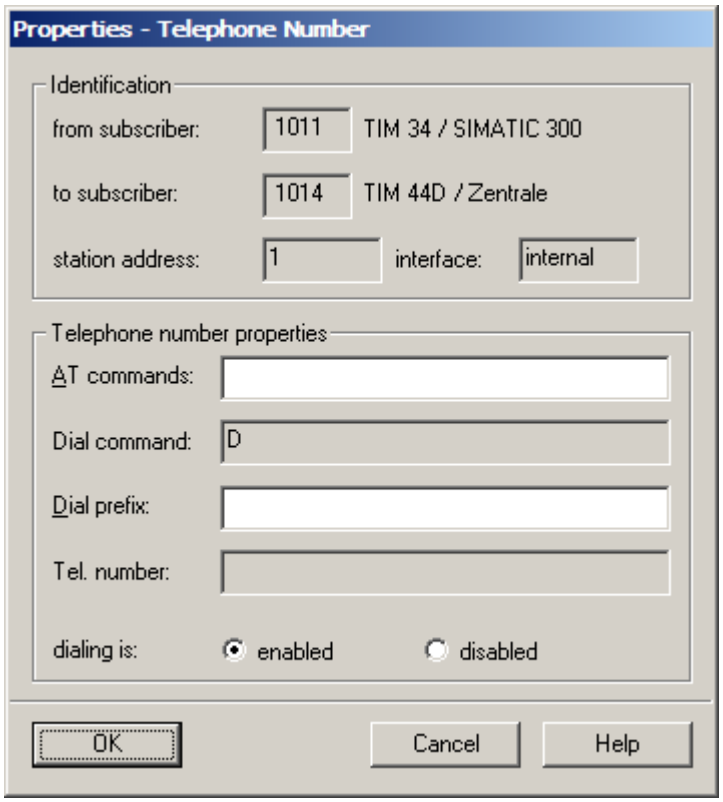


Figure 1-54 *Properties- Telephone Number* dialog

Double-clicking on a subscriber in the list in the *Telephone Directory* tab opens the *Properties - Telephone Number* dialog. This dialog is used when connection-specific modifications to the telephone number are necessary. Examples might be the unlocking of telephone numbers using AT commands or the use of different telephone service providers for connections.

The *Identification* area at the top displays the following information:

- *from subscriber*:
  Source subscriber of the connection
- *to subscriber*:
  Destination subscriber of the connection
- *station address*:
  Station address of the destination subscriber
- *interface*:
  Interface type (internal/external) of the destination subscriber

In the lower *Properties* area, you set the following properties:

- *AT commands*:
  Here, you can set the connection-specific initialization of the modem. The specified AT command is output before the dial command for this number. The AT commands entered must be specified without the *AT* string.
- *Dial command*:
  Displays the configured dial command.
- *Dial prefix*:
  Displays the configured dial prefix. This can be modified.
- *Tel. number*:
  Displays the telephone number of the destination subscriber entered in the network configuration.
- *dialing is:*
  Options: *enabled*, *disabled*
  if the *enabled* option is selected, the dial-up connection is enabled. Otherwise, the dial-up connection is disabled. This function is available only for the TIM 4R-IE.

1.9.9 "DB Configuration" tab

DB Configuration tab

This tab displays the type and number of data blocks required for this CPU type.

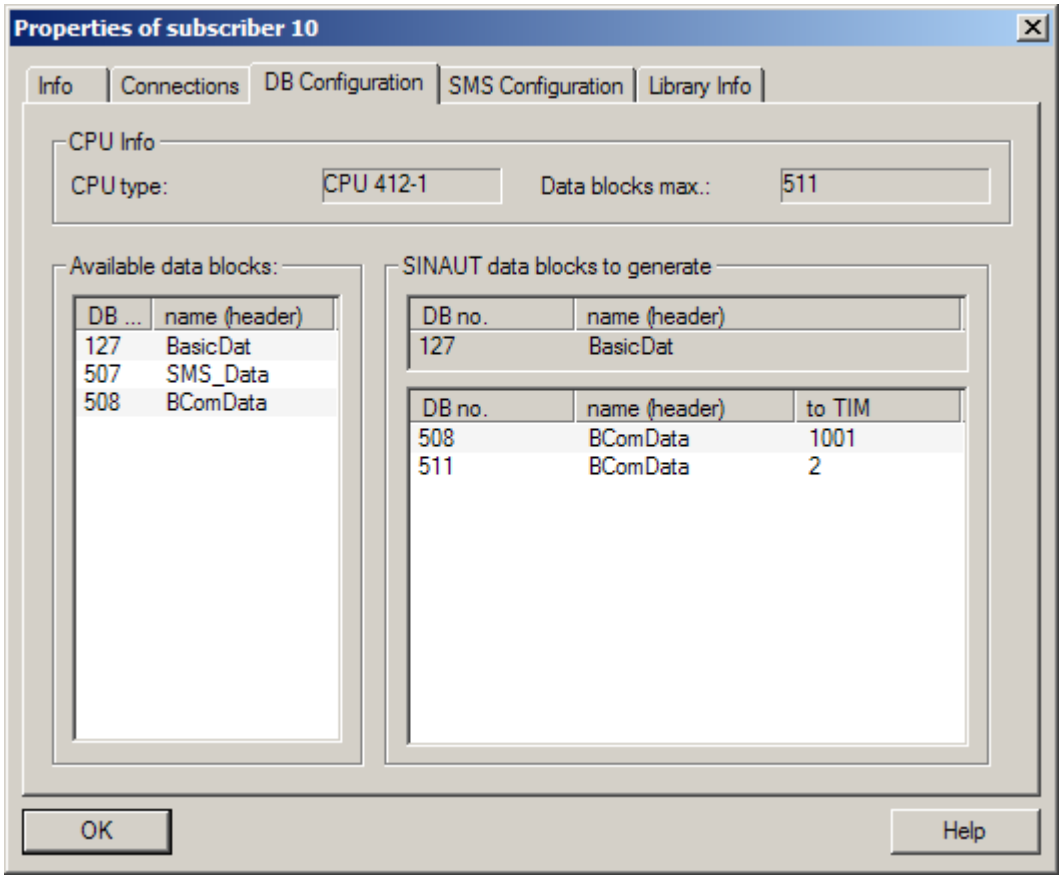


Figure 1-55 *Properties of subscriber* dialog (CPU), *DB Configuration* tab

The following is displayed:

- *CPU type*:
  Type of the current CPU
- *Data blocks max.*:
  Maximum number of data blocks for this CPU (highest DB no.)
- *Available data blocks*:
  DB no. and *name (header)* of the existing data blocks in the offline data management of this CPU. These DB numbers can no longer be assigned.

- *SINAUT data blocks to generate*:
  - *DB no.* and *name (header)* of the DB BasicData:
    This number is either read from the symbol table or is assigned the default 127. The number can only be modified if the complete SINAUT program with the supplied sources is recompiled.
  - *DB no.* and *name (header)* of the communication data blocks and in the column *to TIM* the subscriber number of the partner TIM.
    Based on the existing blocks and the maximum value, the program proposes the numbers for all required data blocks. The user can change these numbers when necessary.

1.9.10 "SMS Configuration" tab

SMS Configuration tab

In the *SMS Configuration* tab, you configure all the SMS messages required for the current CPU. This is only possible if the corresponding DB SMS_Control was created on the CPU. Several SMS data blocks can be defined per CPU and these can in turn contain several SMS messages.

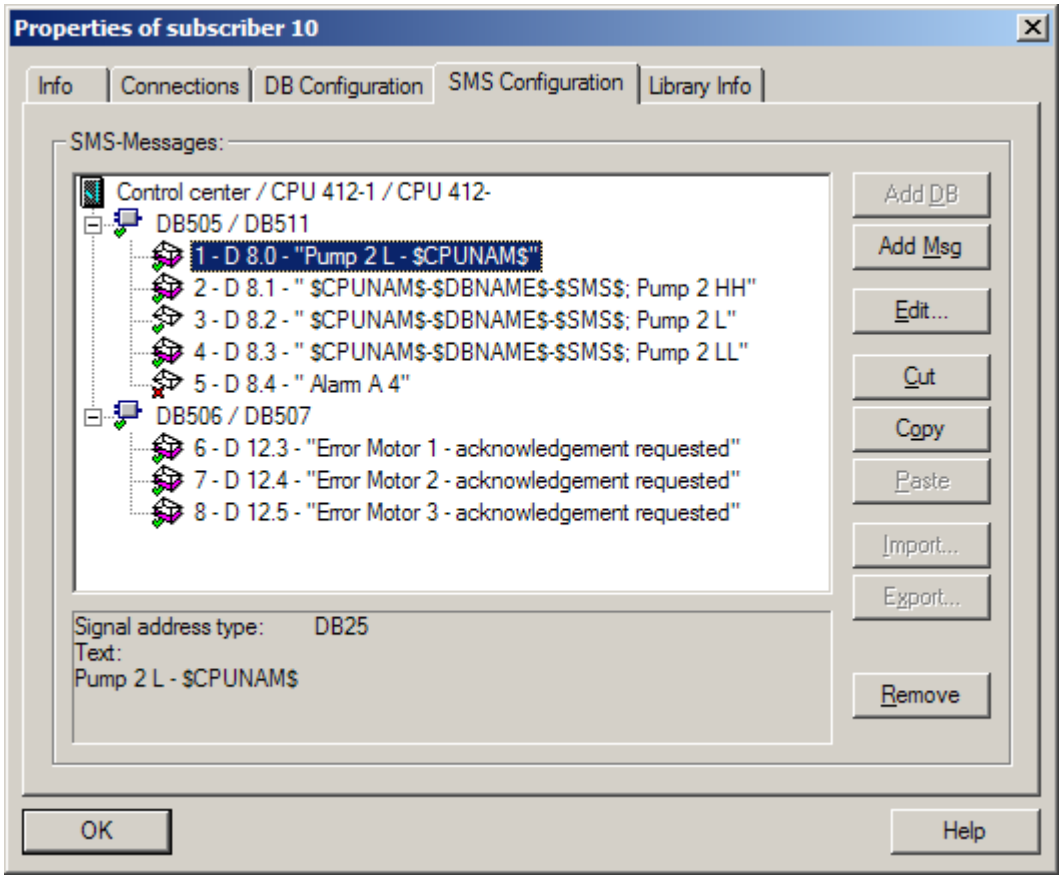

Properties of subscriber 10
Info
Connections
DB Configuration
SMS Configuration
Library Info
SMS-Messages:
Control center / CPU 412-1 / CPU 412-
DB505 / DB511
1 - D 8.0 - "Pump 2 L - $CPUNAM$"
2 - D 8.1 - " $CPUNAM$-$DBNAME$-SMS; Pump 2 HH"
3 - D 8.2 - " $CPUNAM$-$DBNAME$-SMS; Pump 2 L"
4 - D 8.3 - " $CPUNAM$-$DBNAME$-SMS; Pump 2 LL"
5 - D 8.4 - " Alarm A 4"
DB506 / DB507
6 - D 12.3 - "Error Motor 1 - acknowledgement requested"
7 - D 12.4 - "Error Motor 2 - acknowledgement requested"
8 - D 12.5 - "Error Motor 3 - acknowledgement requested"
Signal address type: DB25
Text:
Pump 2 L - $CPUNAM$
Add DB
Add Msg
Edit...
Cut
Copy
Paste
Import...
Export...
Remove
OK
Help

Figure 1-56 *Properties of subscriber* dialog (CPU), *SMS Configuration* tab

In the tree display on the left-hand side, you can see the configured SMS messages in a tree structure with the following information:

- The highest level of the tree displays the current CPU with the text:
  STEP 7 name of the CPU / type of the CPU / CPU short name
- The second level contains the configured data blocks with the text:
  Block number / DB short name
- The third level lists the messages of the relevant DB with the text:
  Message number - signal address (byte.bit) - "message text"

In the tree display, the status of the object is indicated by the type of symbol used.

Table 1- 4 Symbols of SMS Configuration

	Correctly configured DB, data will be generated
	Incompletely configured DB, data will not be generated
	Activated SMS message
	Deactivated or empty SMS message
	Activated SMS message with incoming/outgoing status
	Deactivated or empty SMS message with incoming/outgoing status

Below the tree display there is an information area in which the following characteristic data of the object selected in the list is displayed:

- *CPU* selected:
  Subscriber number and current size of an SMS message
- *DB* selected:
  Number of signals, current DB size, message numbers in this DB, main mobile number and backup mobile number
- *Message* selected:
  Signal address type and message text

To the right of the tree display of the messages, there are buttons that are triggered the available processing functions. The available functions can also be started using the context menu (right mouse button) when an object is selected. Functions are only possible when a CPU, a DB, or a message was selected in the *SMS messages* list.

The range of active functions depends on the selected object. The functions available in SMS configuration are as follows:

- *+ DB* (Add DB) :
  A new data block is added. A free number is searched for as the DB number, starting at the maximum number for the current CPU.
- *+ Message*(Add Msg) :
  A new enter message is added to the current DB. New messages are deactivated as default and must be activated before they can be sent.

- *Edit...* :
  The CPU, a DB or an SMS can be edited. Depending on the selected object, one of the following dialogs opens when the *Edit* function is activated:
  - Edit CPU: The *SMS CPU Configuration* dialog opens.
  - Edit DB: The *SMS DB Configuration* dialog opens.
  - Edit SMS: The *SMS Message Configuration* dialog opens.
- *Cut*:
  The currency selected message is cut and can be inserted again later with the *Paste* function.
- *Copy*:
  The selected message is copied.
- *Paste*:
  The last message to be copied or cut is inserted.
- *Import*:
  A file created with the export function is imported into the selected DB.
- *Export*:
  The data of the current DB is exported to a file. The file format used is the Excel-compatible CSV format; in other words, the exported data can be edited in Excel.
- *Remove*:
  The current object is deleted. With data blocks, you are prompted for confirmation; messages are deleted immediately.

SMS CPU configuration dialog

If you select a station in the *SMS Configuration* tab and click on the *Edit* button, the *SMS CPU configuration* dialog opens. Here, you can see the data of the current CPU that are relevant for SMS configuration.

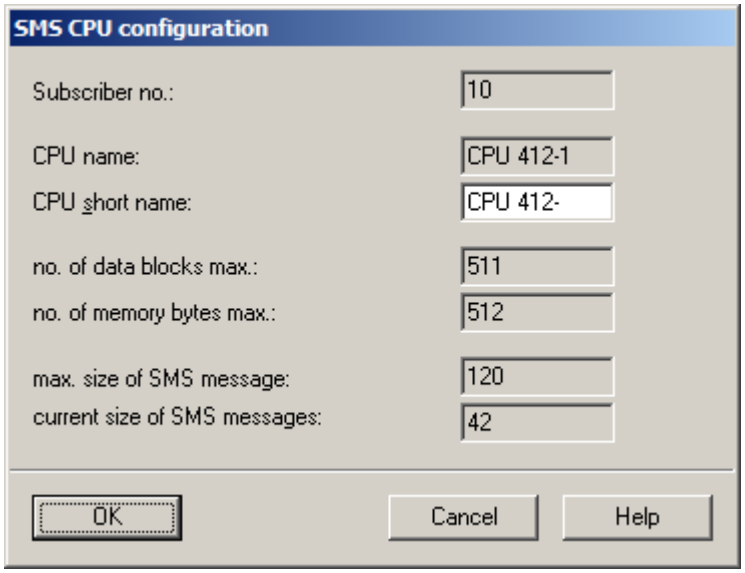


Figure 1-57 *SMS CPU configuration* dialog

For the CPU, you can specify a name with up to eight characters (*CPU short name*). This short name can be inserted with a placeholder string in SMS messages later during *SMS message configuration*. As default, the short name has the first eight characters of the CPU name.

SMS DB data dialog

If you select an SMS DB in the *SMS Configuration* tab and click on the Edit button, the *SMS DB data* dialog opens. This dialog displays and allows you to configure the data of the selected SMS DB.

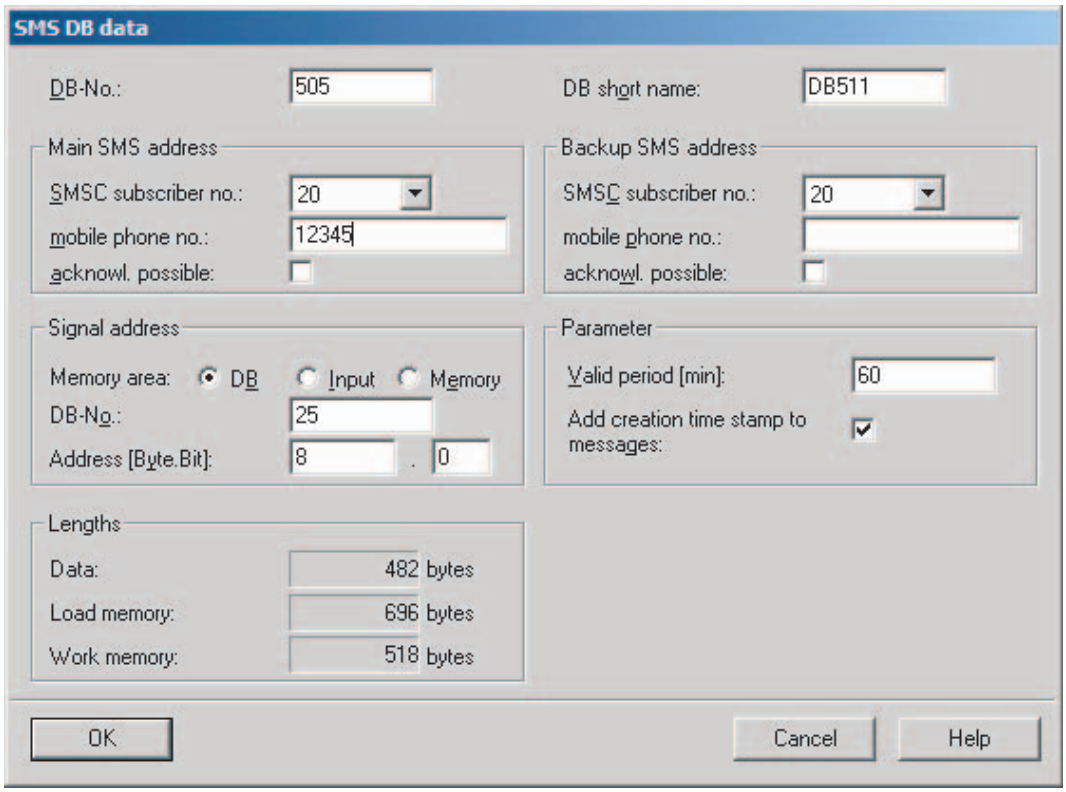


Figure 1-58 *SMS DB data* dialog

The following parameters are available:

- *DB No.*:
  The number used for the SMS DB. The number can be modified, the availability of the specified number on the current CPU is checked. If the number is already assigned by a DB of the CPU user program, an error message is displayed. SMS DBs on the other hand can be overwritten.
- *DB short name*:
  As default, the SMS DB short name has the first eight characters of the CPU name. For the SMS DB, you can specify a DB Short name with up to eight characters. This DB short name can be inserted with a placeholder string in the message text of SMS messages later during *SMS message configuration*.

- In the field *Main SMS address*:
  - *SMSC subscriber no.*: The subscriber number of the SMSC configured in subscriber administration
  - *mobile phone no.*: The mobile phone number to be used for the SMS recipient
  - *acknowl. possible*: The recipient can acknowledge (option selected) or not
- In the field *Backup SMS address* (optional):
  - *SMSC subscriber no.*: The subscriber number of the SMSC configured in subscriber administration
  - *mobile phone no.*: The mobile phone number to be used for the SMS recipient
  - *acknowl. possible*: The recipient can acknowledge (option selected) or not
- In the field *Signal address*:
  The signal address triggers the SMS message. As signal address, bits of a data block (DB), inputs or memory bits can be used. Per SMS data block, the data type of the signals (*address type*), the *DB no.* and the start address within the data field (*address*) must be specified.
  The signals must be located in a contiguous data field, for example in a data block. The first bit triggers the first SMS message, the nth bit triggers the nth SMS message.
  - Address type (DB, Input, Memory)
  - DB no. (only with an addressed type *DB*)
  - Address (Byte.Bit)
- In the field *Parameter*:
  - *Valid period*:
    The period of validity specifies the time within which an SMS message should be delivered and, if selected in the *SMS Message Configuration* dialog, must also be acknowledged. If this has not taken place when the time expires, a diagnostic message is entered in the diagnostic buffer of the CPU.
  - *Add creation time stamp to messages* option:
    In addition to the time added by the SMS center, the creation time of the triggering event can also be sent if this option is selected.
- In the *Lengths* field:
  Here, the length of the DB to be created is calculated based on the currently available *data* and the resulting *Load memory* and the *Work memory requirements*.
  There is no check to make sure that the DB can actually be loaded on the CPU. This is the responsibility of the user.

SMS Message data dialog

When configuring SMS messages, a function is available with which you can check the SMS character set. When entering text for SMS messages, this function blocks certain special characters that can cause problems with some SMS providers.

Before you configure SMS messages for a subscriber, you should decide whether or not this function should be activated. If you activate the function, you can only enter the following characters:
- All numbers
- All letters *except* for umlauts ö, Ö, ü, Ü, ä and Ä
- The special characters *! # % & / ( ) ? * + - . , : ; < = >* and the *blank*.

All other special characters are blocked.
The function is activated in the subscriber administration in the *Extras / Options* menu.

If you select a message in the *SMS Configuration* tab and click on the *Edit* button, the *SMS Message data* dialog opens. You configure an SMS message in this dialog.

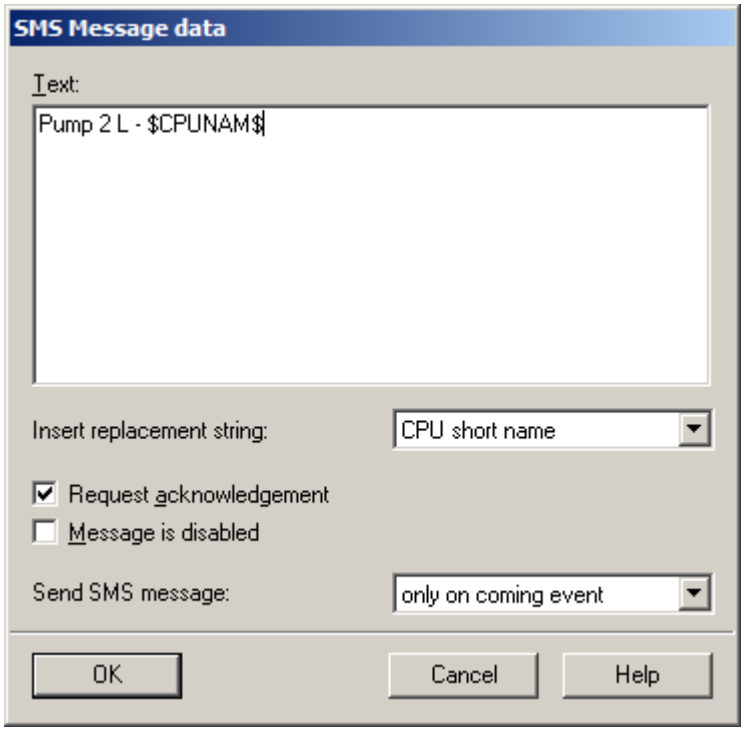


Figure 1-59 *SMS Message data* dialog

The configuration options are available:

- *Text*:
  The message text of the SMS message is entered in the *Text* input box. 120 characters can be entered.
- *Insert replacement string*:
  With this list box, you can insert replacement strings as keywords in the text of the SMS. The following replacement strings can be inserted in the text:
  - CPU short name ($CPUNAM$):
    The CPU short name is configured in the *SMS CPU configuration* dialog
  - DB short name ($DBNAME$):
    The DB short name is configured in the *SMS DB data* dialog
  - Message no. (SMS):
    The message number is a unique number per CPU that is assigned by the SINAUT configuration tool. It can be seen in the tree display of the *SMS messages* list of the *Properties of subscriber* dialog.

  The replacement strings are replaced by the actual values when the SMS is generated. By specifying the 3 strings, every SMS message in an S7 project can be uniquely identified.
- *Request acknowledgment* option:
  You can specify whether or not an acknowledgment is required for this message.

- *Message is disabled* option:
  If message output is activated, this option must be deselected (no check mark).
- *Send SMS message*:
  The list box indicates whether the message is sent on an event entering state (coming) or an event leaving state (going).

1.9.11 "MSC Station List"

Requirement for configuring the MSC subscribers

Before you configure the MSC subscribers in the properties dialog of the MSC master, you have already performed the steps listed below:

- Networking of the interface of the TIM module and assignment of the network node type MSC master, MSC node station or MSC station
- Configuring Internet providers

You will find an configuration overview of "simple Internet communication" in the section GPRS / Internet communication: Overview of configuration (Page 15).

Information in the "MSC Station List" tab

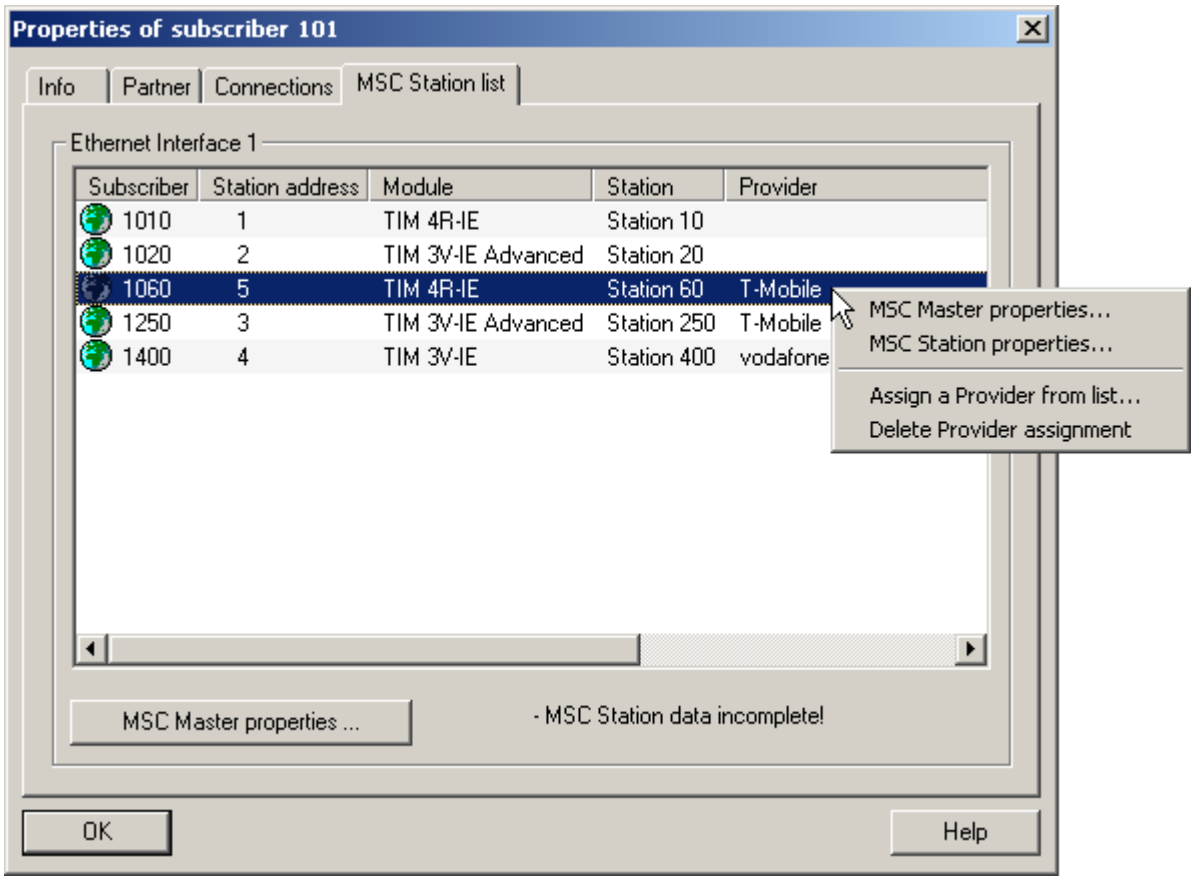


Figure 1-60 "Properties of Subscriber" dialog of the MSC master, "MSC Station List" tab

The "Ethernet interface 1" table shows all MSC stations of the project and specifies the subscriber number, station address, module type, station name (from NetPro) and provider name. The globe icon at the start of the row shows the status of the GPRS provider.

The globe icon in this dialog has a the following meaning depending on its color:

- Green ( )

  All the required properties of the provider are configured.

- Black and white ( )

  The configured data of the provider is not complete.

  If you select subscribers that have not been fully configured, the incomplete data is shown in the lower part of the dialog.

Configuration in the "MSC Station List" tab

Follow the steps below to configure the MSC subscribers:

1. Configure the data of the MSC master by clicking the "MSC Master Properties..." button.
2. Configure the data of the MSC stations by double-clicking on a station in the list or using the "MSC Station Properties" shortcut menu command.

3. Assign a GPRS provider to the MSC stations (with MD720-3) by selecting a station in the list and then the shortcut menu command "Assign Provider...". You can select one or more stations at the same time.

   You can delete the provider assignment again with the shortcut menu command "Delete Provider Assignment".

   Here, you cannot assign a provider to stations with a DSL connection. You make the assignment in the configuration of the relevant DSL modem.
4. Close the properties dialog with "OK" when you have completed the MSC configuration.

Configuring the MSC master

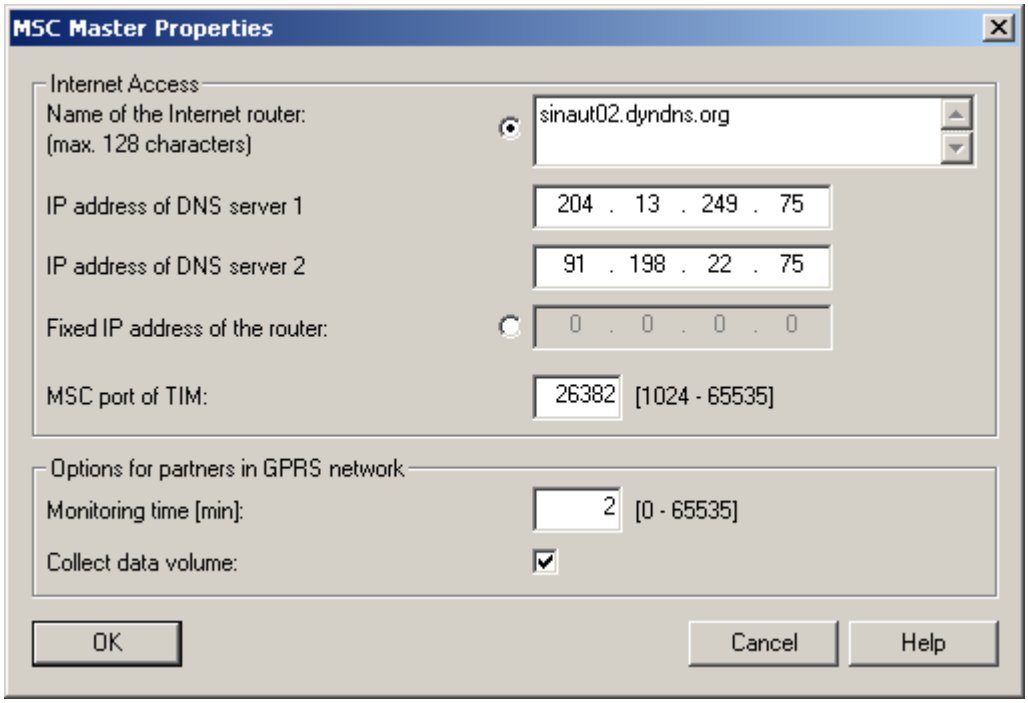


Figure 1-61 "MSC Master Properties" dialog

In the "Internet Access" box, you configure the address of the Internet router via which the MSC master is connected to the Internet and the port of the MSC master used for MSC communication. You can specify the Internet router either with a DNS name or a fixed IP address.

- Name of the Internet router:

  Enable this option if you enter the DNS name of the router.

  – Enter the name of the router in the input box.

    Max. 128 characters. The following characters are permitted: a...z, A...Z, 0...9, comma (,), period (.), hyphen (-), underscore (_) and the blank

  – Enter the IP address of at least one DNS server.

- Fixed IP address of the router:

  Enable this option if you configure the router with a fixed IP address.

  Enter the IP address of the TIM interface that is connected to the Internet or the interface of the router if you have configured a router (option "Use router" in the "Parameters" tab of the properties dialog of the Ethernet network node of the TIM).

  As default, the IP address is displayed from the interface parameter assignment of the TIM.

- MSC port of TIM

  If necessary, assign the number of the MSC port (no. 1 024 ... 65 535 can be selected). As default, port 26 382 is used.

In the "Options for partners in GPRS network" box, you can make the settings for GPRS stations (with MD720-3).

- Monitoring time [min]

  The monitoring time in minutes is used to monitor the MSC tunnel in the GPRS network. When using a GPRS station, if there is no data traffic between the MSC master and GPRS station within the monitoring time, the MSC master sends a monitoring message to the GPRS station. If there is no sign of life from the MSC master during the monitoring time, the master station and all subscribers that can be reached via the MSC master are indicated as disrupted in the GPRS station. En entry is made in the diagnostics buffer of the TIM in the station and the MD720-3 is reinitialized.

  The entry "0" means: No monitoring.

- Collect data volume

  If this option is enabled (checkmark), the master TIM makes the transferred data of all GPRS stations available.

  The data is available in the SINAUT diagnostics and service tool, see TIM Diagnostics (Page 345).

Configuring an MSC station/node station

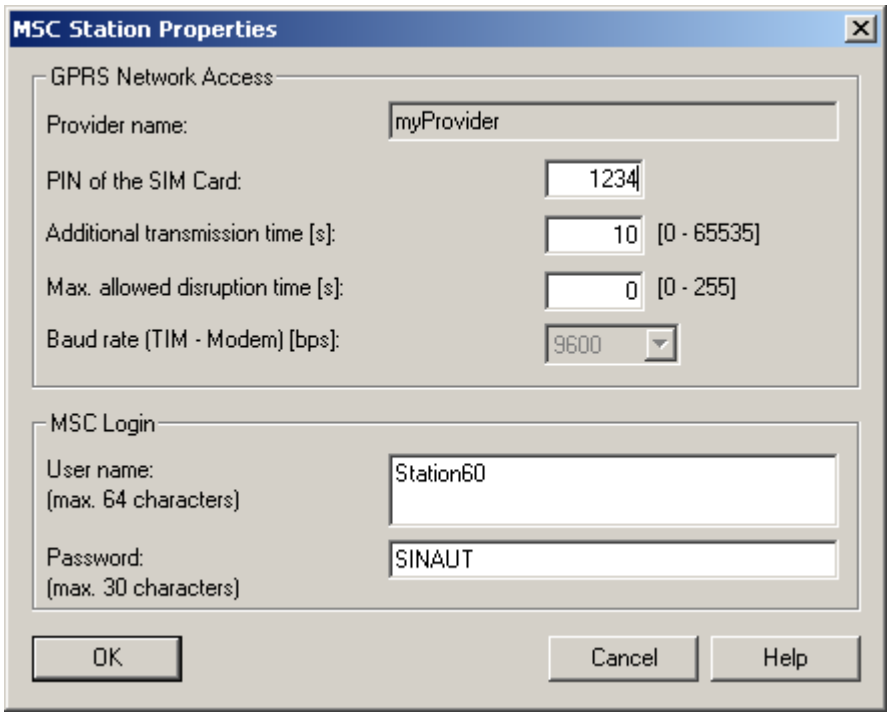


Figure 1-62 "MSC Station Properties" dialog

In this dialog, you configure the access data for the connected network.

"GPRS Network Access" box

In this box, you configure the Internet access data for GPRS stations (with MD720-3). For stations with a DSL connection, you configure the relevant data on the DSL router.

- Provider name

  If you have not yet assigned a GPRS provider,the first provider from the list of GPRS providers is entered as default. You can assign the provider in the MSC station list, see section "MSC Station List" (Page 110).

- PIN of the SIM card

  Here, you enter the PIN of the SIM card for the MD720-3.

- Additional transmission time [s]

  This is an offset added to the transmit retry time. The transmit retry time is calculated automatically on the TIM. The character delay time is also calculated from the "additional transmission time" parameter (character delay time = additional transmission time divided by 5).

  For GPRS networks, a value of 10 seconds as in the default setting is normally adequate.

- Max. allowed disruption time [s]

  Here, you can enter the tolerance time for connection disruption detected by the TIM. If there is still a disruption on the connection when the set time has elapsed, the disruption is signaled to all connection partners of the disrupted station.

  If networks are regularly disrupted, it may be useful to increase the default value of 10 seconds. Increasing the allowed disruption time delays signaling of station failures and so reduces the number of organizational messages when stations return.

- Baud rate (TIM - modem) [bps]

  This is the speed at which the TIM and modem communicate. The default baud rate is 9 600 bps and cannot be changed.

"MSC Login" box

Here, you configure the two mandatory boxes "User name" and "Password" for all MSC stations/node stations with GPRS or DSL connection. The data is required for establishing the connection and logging on with the MSC master.

- User name

  Enter the user name here with a maximum of 64 characters.

  The following characters are permitted: a...z, A...Z, 0...9, comma (,), period (.), hyphen (-) and underscore (_)

- Password

  Enter the password here with a maximum of 30 characters.

  The following characters are permitted: a...z, A...Z, 0...9, comma (,), period (.), hyphen (-) and underscore (_)

The "Provider Assignment" dialog

You open the "Provider Assignment" dialog from the properties dialog of the MSC station in the SINAUT configuration tool. You select an MSC station (with MD720-3) in the "MSC Station List" tab and then select the "Assign provider..." shortcut menu command (right mouse button). The "List of GPRS Providers" dialog opens.

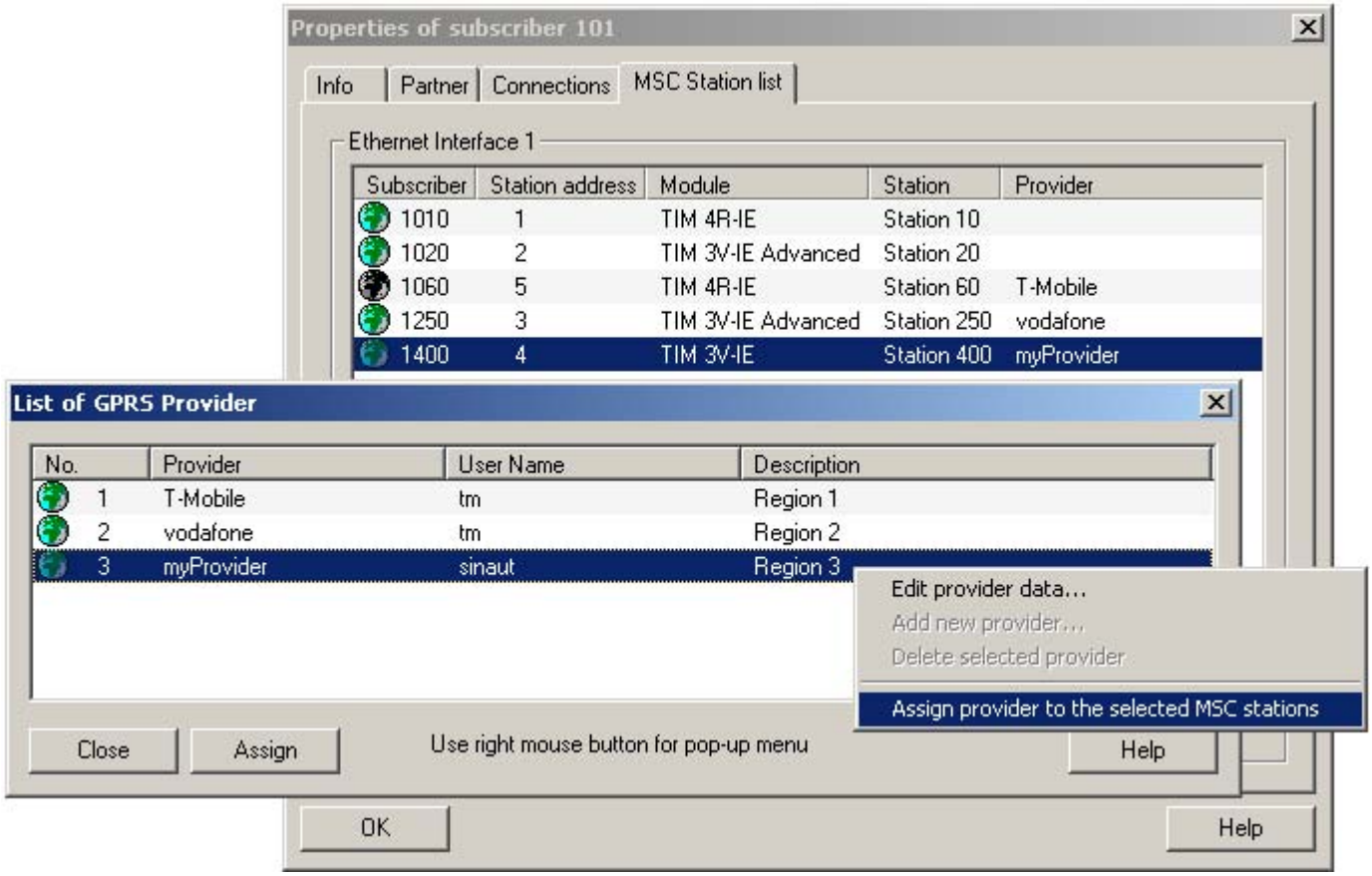


Figure 1-63 "List of GPRS Providers" dialog opened from the "MSC Station List" tab

The dialog box lists the Internet providers configured for MSC communication showing the user name, a description and the globe icon.

The globe icon in this dialog has a the following meaning depending on its color:

- Green ( )

  All the required properties of the provider are configured.

- Black and white ( )

  The configured data of the provider is not complete.

You cannot assign incompletely configured providers.

Assigning providers and saving the assignment

Select a provider from the list and then select the "Assign provider to the selected MSC stations" shortcut command (right mouse button).

The assignment of the provider is saved when you exit the properties dialog in the SINAUT configuration tool ("MSC Station List" tab) with "OK".

Configuring the provider data

You cannot edit the provider data at this point in the "List of GPRS Providers" dialog. To edit the data, close both open dialogs and reopen the "List of GPRS Providers" dialog again by clicking on the globe icon ( ) in the toolbar in the window of the SINAUT configuration tool.

1.9.12 "Library Info" tab

Library Info tab

This tab shows the name, path, version, date of creation and source information for the German and English version of the *SINAUT TD7 library* with which system data for the current CPU will be generated.

1.9.13 Printing subscriber lists

To provide users with an overview of the existing SINAUT subscribers, the SINAUT configuration tool provides the option of printing out the list of subscribers.

A print preview function is available to check the output before printing.

1.10 TD7onTIM software package

1.10.1 Introduction

SINAUT communication between the CPU modules or between CPU modules and a control center is implemented with the aid of TIM modules. The organization of SINAUT communication is handled by the *SINAUT TD7* software package. In a SINAUT station, the TD7 software must be available either on the CPU or on a TIM. For these two situations, the SINAUT TD7 software is available in two variants:

- **TD7onCPU**:
  The TD7onCPU software package is used on the CPU and is mandatory in all stations with TIM modules that do *not* have TD7onTIM.
  In stations *with* TD7-compliant TIM modules, it *can* be used on the CPU modules. You configure in the STEP 7 Editor in STL (Statement list), FBD (Function Block Diagram) or LAD (Ladder Logic).
  TD7onCPU is described in a separate section.
- **TD7onTIM**:
  The TD7onTIM software package is available only for TIM modules that are TD7onTIM-compliant (for example TIM 3V-IE) and is part of the TIM firmware.
  If you use TD7onTIM, little or no work memory is required on the CPU. You configure in the subscriber administration of the SINAUT ST7 configuration tool.

Note

TD7onTIM cannot be used if the TIM sends or receives ST1 messages. In this case, you must use TD7onCPU

This section deals only with configuration of *TD7onTIM*.

If the SINAUT project you are editing does not contain TD7onTIM-compliant TIM modules, you can skip this section and continue configuration with saving and generating system data.

If the project you are editing contains TD7onTIM-compliant TIM modules, on which you want to configure the TD7onTIM software, you do this in the subscriber administration of the SINAUT ST7 configuration tool.

1.10.2 Basic functions and components of TD7onTIM

Basic functions and components of TD7onTIM

TD7onTIM handles the sending and receiving of process data for the local CPU. Data to be sent by TD7onTIM, is read by the TIM over the backplane bus of the CPU, received data is written to the CPU.

SINAUT communication makes use of SINAUT objects: Which data is to be sent or received can be configured over standardized data objects. These are collected in the TD7onTIM standard library. This library also contains the system objects with which system information is displayed and with which system functions are activated and set.

The parameter assignment for TD7onTIM is made in the subscriber administration of the SINAUT ST7 configuration tool in the following steps:

- Setting basic parameters for TD7onTIM
- Specifying the parameters specific to the destination subscribers
- Inserting system objects and assigning parameters to them
- Inserting data objects including their send and receive channels and assigning parameters to them

The basic settings for TIM subscribers with TD7onTIM

For each TIM with TD7onTIM, several settings must be made that are always required when working with TD7onTIM, for example specifying the read/write cycle. The settings are made in the basic settings for TIM subscribers with TD7onTIM.

The parameters specific to the destination subscribers

Each TIM with TD7onTIM can exchange data with one or more partners, known as destination subscribers. Which subscribers in the project are suitable as destination subscribers depends on the connection configuration. Here, you specify which TIM with TD7onTIM will have a connection with which SINAUT subscriber in the network.

Several settings are required for each of the possible destination subscribers that apply to data traffic between the TD7onTIM of a project and this subscriber, for example whether the

subscriber expects data messages with a time stamp. This information is specified in the parameters specific to destination subscribers.

The system objects

The system objects provide system-relevant information for the CPU user program. Configuration of the system objects is optional.

- The WatchDog:
  The WatchDog indicates to the CPU program whether the communication between CPU and local TIM is still working; in other words, whether TD7onTIM is still reading from and writing to the memory areas of the CPU.
- The PartnerStatus:
  The PartnerStatus indicates to the CPU program whether communication with its partners (other ST7 CPUs or ST7cc/sc control centers) is OK or disrupted.
- The OpInputMonitor:
  This indicates the status of operating input to the CPU program (with command, setpoint, and parameter input).

The data objects

The sending and receiving of process data is configured with the aid of standardized data objects. According to the two transmission directions, these are divided into:

- Data objects for acquiring and sending data,
  Their names have the ending *_S* for Send.
- Data objects for receiving and outputting data,
  Their names have the ending *_R* for Receive.

In terms of the names, the data objects of TD7onTIM are identical with those of the data objects of TD7onCPU. In terms of functionality, they are compatible with each other; in other words, communication between data objects of TD7onTIM and the corresponding data objects of TD7onCPU is guaranteed.

The data objects are available in a standard library and are inserted from the library into the TD7onTIM configuration. An example of a data object is *Ana04W_S* that organizes the transmission of 4 analog values.

Each data object contains one or more send or receive channels. The number and type of send and receive channels per data object cannot be modified.

The send and receive channels

The send and receive channels of the data objects are responsible for the processing of an individual process value, for example for processing and sending an analog value or receiving and outputting a message byte.

The data object *Ana04W_S*, for example, has 4 send channels of the type *send analog value*.

1.10.3 Parameter assignment dialogs for TD7onTIM

Calling and appearance of the parameter assignment dialogs

Information on displaying and assigning parameters with the TD7onTIM software is divided into three areas of subscriber administration:

- The *directory tree* in the left-hand part of subscriber administration
- The *list box* at the top right
- The *parameter assignment window* below the list box

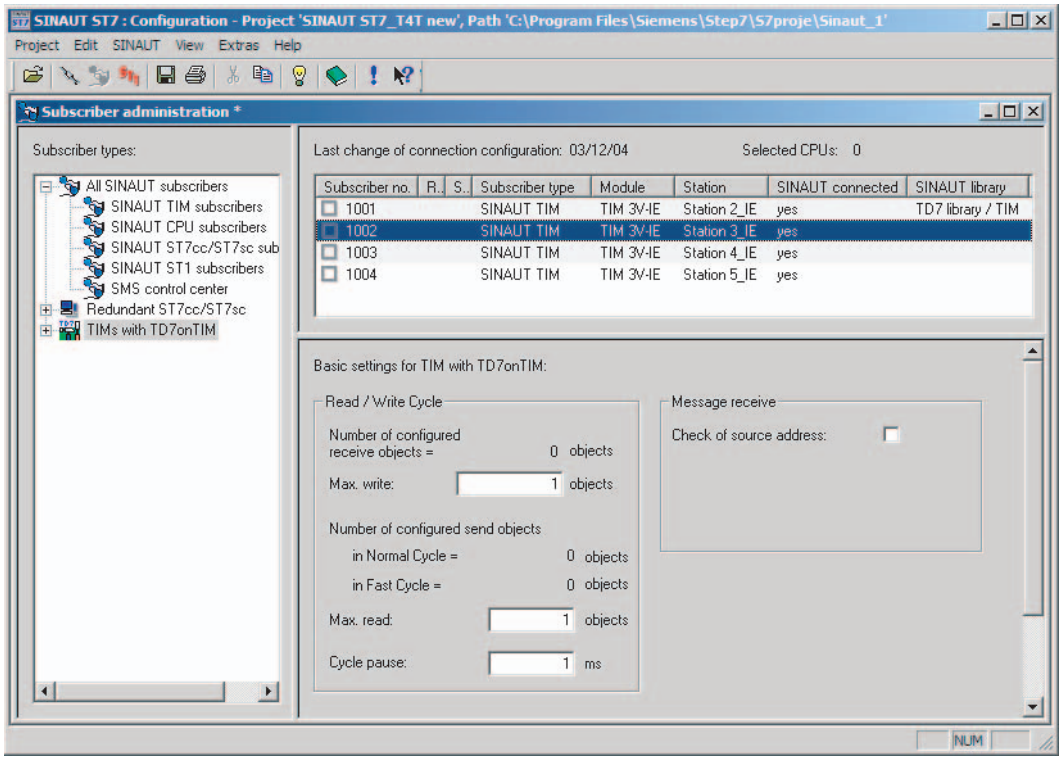

SINAUT ST7 : Configuration - Project 'SINAUT ST7_T4T new', Path 'C:\Program Files\Siemens\Step7\S7proje\Sinaut_1'
Project Edit SINAUT View Extras Help
Subscriber administration *
Subscriber types:
All SINAUT subscribers
SINAUT TIM subscribers
SINAUT CPU subscribers
SINAUT ST7cc/ST7sc sub
SINAUT ST1 subscribers
SMS control center
Redundant ST7cc/ST7sc
TIMs with TD7onTIM
Last change of connection configuration: 03/12/04
Selected CPUs: 0
Subscriber no. R.. S.. Subscriber type Module Station SINAUT connected SINAUT library
1001 SINAUT TIM TIM 3V-IE Station 2_IE yes TD7 library / TIM
1002 SINAUT TIM TIM 3V-IE Station 3_IE yes
1003 SINAUT TIM TIM 3V-IE Station 4_IE yes
1004 SINAUT TIM TIM 3V-IE Station 5_IE yes
Basic settings for TIM with TD7onTIM:
Read / Write Cycle
Number of configured receive objects = 0 objects
Max. write: 1 objects
Number of configured send objects
in Normal Cycle = 0 objects
in Fast Cycle = 0 objects
Max. read: 1 objects
Cycle pause: 1 ms
Message receive
Check of source address:
NUM

Figure 1-64 SINAUT ST7 subscriber administration with the *TIMs with TD7onTIM* directory selected, the list box and the parameter assignment window of the *basic settings for TIM with TD7onTIM*

You can change the size of the three windows in subscriber administration. The vertical and horizontal divisions between the windows can be moved with the mouse.

The directory tree

The TD7-compliant TIM modules of a project are shown in the directory tree in the *TIMs with TD7onTIM* directory. If you expand the directory with the (+) symbol or double-click on the directory, the following contents are displayed:

- The *All Destination Subscribers* directory
- The directories of all TIMs with TD7onTIM

If you expand a single TIM directory, the *data objects* already configured on this TIM are displayed.

The list box

At the top right of the subscriber administration there is a list box that lists certain subscriber types, objects, or send/receive channels depending on what is selected in the directory tree. By successively expanding the *TIMs with TD7onTIM* directory, the list box displays the following content:

- The TD7-compliant TIMs of the project
- The destination subscribers with which the TD7-compliant TIMs can communicate
- The system and data objects of a TD7-compliant TIM
- The channels of a data object

The SINAUT objects are displayed with the following symbols:

- Blue symbols: System objects
- Yellow symbols: Data objects

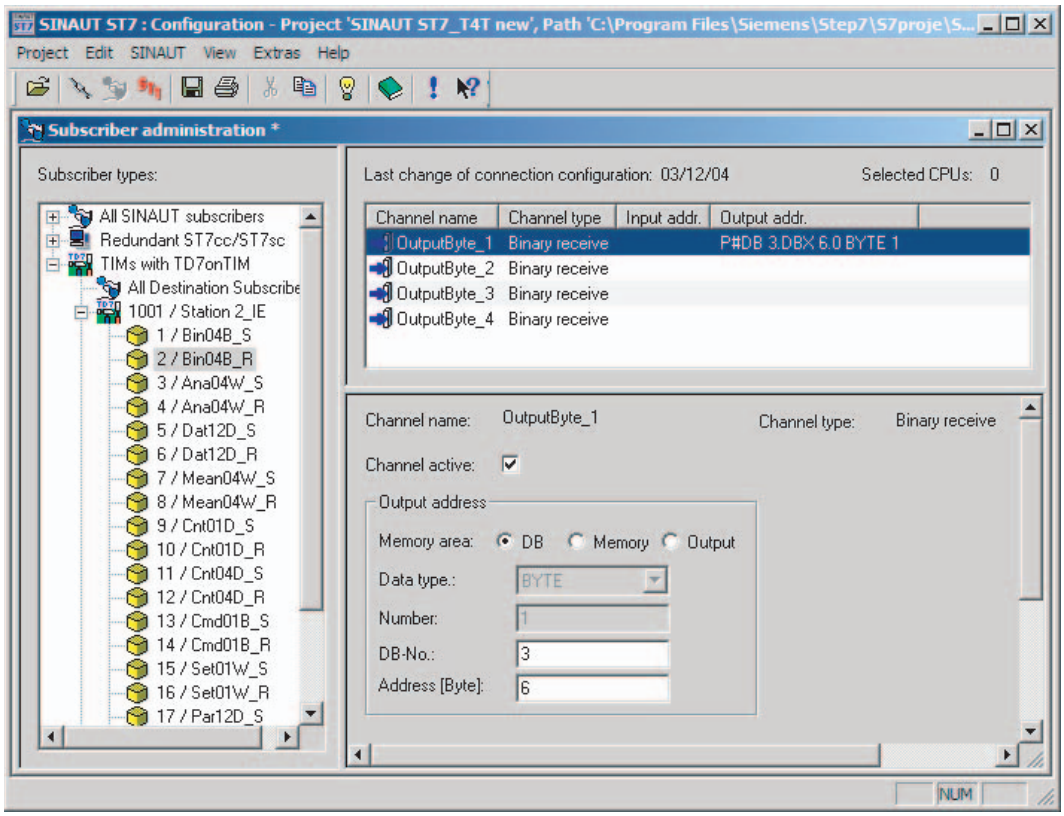


Figure 1-65 Selected data object *Bin04B_R* with selected channel in the list box and the parameter assignment window of a receive channel

The parameter assignment windows

In the parameter assignment windows below the list box, you set the parameters for the *TD7onTIM*-relevant subscribers, objects and channels. Depending on what is selected in the directory tree or in the list box, the following dialogs are displayed:

- Parameter assignment dialog of the basic parameters for a TD7-compliant TIM
- Parameter assignment dialog of a destination subscriber
- Parameter assignment dialog of a system or data object
- Parameter assignment dialog of a send or receive channel
  - Send channels are displayed with an outgoing arrow.
  - Receive channels are displayed with an incoming arrow.

To open a parameter assignment dialog, select the relevant subscriber, object or channel in the *list box*.

If a subscriber or object is selected in the *directory tree*, the parameter assignment dialog of the first object of the subscriber or the first channel of the data object is displayed.

Parameter entries

To simplify data entry, the parameter assignment dialogs are not opened or closed using separate buttons or menus but are displayed automatically when a subscriber, object or channel is selected with the mouse in the list box above.

The entries in the parameter assignment dialogs are not applied using a separate button, but immediately:

- When you activate or deactivate an option
- After entering data when you exit the input box with the mouse or tab key

The entries are applied permanently using the *Save* function.

1.10.4 Basic settings for TIM subscribers with TD7onTIM

To make the basic settings for the TD7 software of the individual TIM subscribers, you first select the *TIMs with TD7onTIM* directory in the directory tree. The list box then displays all the TD7-compliant TIM modules of the project.

Below the list box, the parameter assignment dialog appears for the TIM selected either automatically or with the mouse.

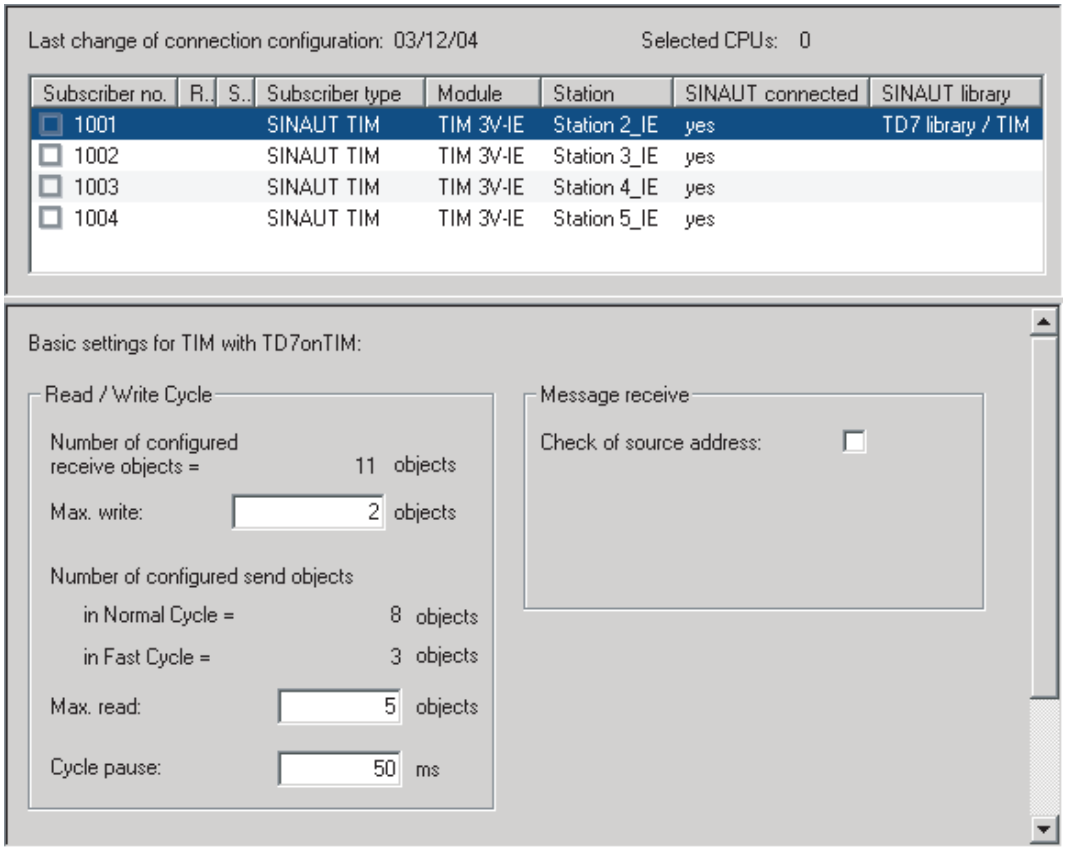


Figure 1-66 Parameter assignment dialog for basic settings for TIM subscribers with TD7onTIM

The parameters of the basic settings for TIM subscribers with TD7onTIM relate to

- The configuration of the read/write cycle of the TIM and
- Checking the source address when a message is received

The read/write cycle

Data to be sent by TD7onTIM, is read by the TIM over the backplane bus of the CPU and received data is written to the CPU.

The TIM also writes system information to the CPU (see system objects *Watchdog*, *PartnerStatus* and *OpInputMonitor*) and certain data is reset; in other words, 0 is written. In the latter situation, this involves send trigger and command information that was read from the memory bit area or data blocks. TD7onTIM ensures that these are reset to 0 automatically after they have been acquired. All of these procedures take place within a defined and selectable read/write cycle.

The writing and reading of data takes place in consecutive read/write cycles. A basic cycle of the read/write cycle of TD7onTIM is made up as follows:

1. Write all pending system information (see system objects) to the CPU and reset all currently acquired send triggers and commanded entries. If no such data is currently pending, there is no write procedure in the basic cycle.
2. Read *all* data of the send objects that were assigned to the *fast cycle*. If no objects were configured for the *fast cycle*, this read procedure is omitted in the basic cycle.
3. Read data from some of the send objects that were assigned to the *normal cycle*. How many objects per basic cycle will be read can be set by the user. Refer to the *Max. read* parameter below.
4. Write data of some of the currently pending receive objects.
   How many objects this can be as a maximum per basic cycle can be selected by the user. Refer to the Max. write parameter below.
   If less received data is currently pending than his permitted as maximum per basic cycle, only this subset is written in the basic cycle.
   If there are currently no received data from the remote partner, this write procedure is omitted in the basic cycle.
5. Cycle pause (optional) to relieve the TIM and backplane bus communication.

With the *Max. read* and *Max. write* parameters and by specifying how many objects are assigned to the fast cycle, the user can determine the duration of a basic cycle. Essentially, this specifies how fast the fast cycle really is: It is identical with the duration of the basic cycle. With the default 1 for the *Max. read* and *Max. write* parameters, the basic cycle has the shortest possible duration.

It must also be taken into account that the make-up of the basic cycle decides how long TD7onTIM requires to read all the data of the objects assigned to the normal cycle once. If, for example, 12 objects are assigned to the *normal cycle* and if *Max. read* is set to 2 objects per basic cycle, it takes 6 basic cycle is until all the data of the 12 objects have been read once completely from the memory areas of the CPU.

Parameters in the *read/write cycle* field

Name:	**Max. write**
Range of values:	1 ... 32000
Default:	1
Explanation:	This is the maximum number of (different) data objects whose data is written to the CPU per basic cycle. If there are several messages of the same receive object in the buffer, only the data of one message of this object is written per basic cycle. As information, the number of receive objects configured for the TIM in total by the user is displayed above the input box beside *Number of configured receive objects*.

Name:	**Max. read**
Range of values:	0 ... 32000

Default:	1
Explanation:	This is the maximum number of data objects whose data is read from the CPU per basic cycle. As information, above the input field beside *Number of configured send objects*, you can see how many send objects the user configured for the TIM - in the *normal cycle* and - in the *fast cycle* .

Name:	**Cycle pause**
Range of values:	0 ... 32000 [ms]
Default:	1
Explanation:	This parameter specifies the duration of an optional pause between 2 basic cycles. A pause may be necessary if communication of other modules on the backplane bus is disrupted too much by the write and read jobs between the TIM and CPU. This also applies to subscribers on the MPI bus (further CPUs or PG) if the backplane bus is implemented as a party line. By setting a suitable time for the pause, the other bus subscribers have time for their communication. Specifying a cycle pause may also be necessary to relieve the TIM itself; in other words, when it becomes clear that it has too little time for other tasks due to the fast read/write cycle.

Parameters in the *Message receive* area

Name:	**Check of source address**
Range of values:	Function active, function deactivated
Default:	Function active
Explanation:	With this parameter, you specify whether or not the source address of the sending subscriber is checked prior to accepting data from a received message. If the function is activated, all messages that do not originate from the configured partner are discarded. **Note**: If a data object receives messages from several partners, the check of the source address must be deactivated.

Copying basic settings to other TIMs

Once the basic settings of a subscriber with TD7onTIM have been made, the settings can also be transferred to other TIMs that require the same parameters. Follow the steps outlined below:

1. Select a TIM for which you have already set the parameters in the list box.
2. Select *Copy* in the context menu (right mouse button).
3. Select a second TIM.
4. Transfer the parameters to this second TIM using *Paste* in the context menu.

1.10.5 Subscriber-specific parameters of TD7onTIM

Each TIM with TD7onTIM can exchange data with one or more partners, known as destination subscribers. Which subscribers in the project are suitable as destination subscribers depends on the connection configuration. Here, you specify which TIM with TD7onTIM will have a connection with which SINAUT subscriber in the network.

Several settings are required for each of the possible destination subscribers that apply to data traffic between the TD7onTIM of a project and this subscriber, for example whether the subscriber expects data messages with a time stamp.

To set the parameters specific to destination subscribers, open the *TD7 on TIM* directory and select the *All Destination Subscribers* directory. The list displays all potential destination subscribers of the TD7-compliant TIM modules. These are:

- SIMATIC S7 CPU modules
- SINAUT ST7cc/sc control centers

Below the list box, the parameter assignment dialog of the destination subscriber (selected automatically or with the mouse) opens.

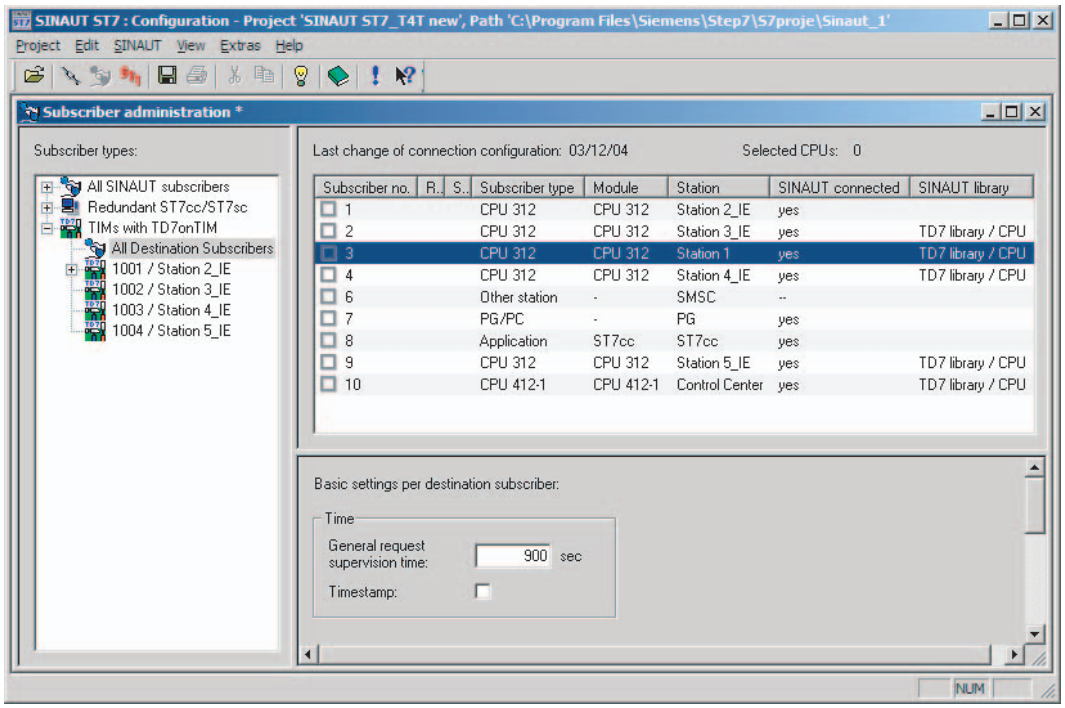

SINAUT ST7 : Configuration - Project 'SINAUT ST7_T4T new', Path 'C:\Program Files\Siemens\Step7\S7proje\Sinaut_1'
Project Edit SINAUT View Extras Help
Subscriber administration *
Subscriber types:
All SINAUT subscribers
Redundant ST7cc/ST7sc
TIMs with TD7onTIM
All Destination Subscribers
1001 / Station 2_IE
1002 / Station 3_IE
1003 / Station 4_IE
1004 / Station 5_IE
Last change of connection configuration: 03/12/04
Selected CPUs: 0
Subscriber no. R.. S.. Subscriber type Module Station SINAUT connected SINAUT library
1 CPU 312 CPU 312 Station 2_IE yes
2 CPU 312 CPU 312 Station 3_IE yes TD7 library / CPU
3 CPU 312 CPU 312 Station 1 yes TD7 library / CPU
4 CPU 312 CPU 312 Station 4_IE yes TD7 library / CPU
6 Other station - SMSC --
7 PG/PC - PG yes
8 Application ST7cc ST7cc yes
9 CPU 312 CPU 312 Station 5_IE yes TD7 library / CPU
10 CPU 412-1 CPU 412-1 Control Center yes TD7 library / CPU
Basic settings per destination subscriber:
Time
General request supervision time:
900
sec
Timestamp:
NUM

Figure 1-67 Directory tree, list box, and parameter assignment dialog of the destination subscriber-specific parameters of TD7onTIM for a destination subscriber

The parameters to be entered here are valid in terms of communication with the configured communication partners (destination subscribers) for all TD7-compliant TIM modules of the project.

The parameter assignment dialog for the destination subscriber-specific parameters is opened and the parameters are set for every configured destination subscriber.

The following destination subscriber-specific parameters are available:

Name:	**General request supervision time**
Range of values:	10 ... 32000 s
Default:	900 s
Explanation:	The *General request supervision time* is the maximum time required by a destination subscriber to respond fully to a general request (GR). If the GR response has not arrived completely at the requesting TIM module when the supervision time has expired, a message is entered in the diagnostic buffer of this TIM module and an ID in the objects of the TD7onTIM involved. A TIM can only send requests to SINAUT stations with TD7 software; in other words, a CPU with TD7onCPU or a CPU with a local TIM with TD7onTIM. The time should be set generously. Particularly with dial-up connections, remember that the time for connection establishment is also included in the supervision time. In addition to this, on dial-up stations with messages stored in the send buffer, the GR can be further delayed because the requested messages are entered after all other messages in the send buffer.

Name:	**Time stamp**
Range of values:	Function active, function deactivated
Default:	Function deactivated
Explanation:	This parameter specifies whether or not messages with a time stamp are sent to this destination subscriber. If this is the case, the *Timestamp* option must be activated. TD7onTIM can send either all messages with or all messages without a time stamp to a destination subscriber. Mixing messages to a destination subscriber with and without time stamps is not possible.

Copying parameters to other destination subscribers

Once the destination subscriber-specific parameters have been set for a destination subscriber, they can be transferred to other destination subscribers that require the same parameters. Follow the steps outlined below:

1. Select a destination subscriber for which you have already set the parameters in the list box.
2. Select *Copy* in the context menu (right mouse button).
3. Select a second destination subscriber in the list box.
4. Transfer the parameters to this second destination subscriber using *Paste* in the context menu.

1.10.6 Configuring SINAUT objects

Open in the standard library of TD7onTIM

After setting the basic parameters of TD7onTIM and the destination subscriber-specific parameters, the SINAUT objects of TD7onTIM are configured. To do this, a TIM module is selected in the directory tree. If no SINAUT objects have yet been configured for the TIM (as is the case in a new project), the directory of the TIM cannot be expanded any further and the list box is empty.

The standard library can be opened either using:

- The *Standard library* button in the toolbar
- The *SINAUT / Standard Library for the TIM* menu
- The F7 function key

Every TIM with TD7onTIM now has the required SINAUT objects added from the standard library.

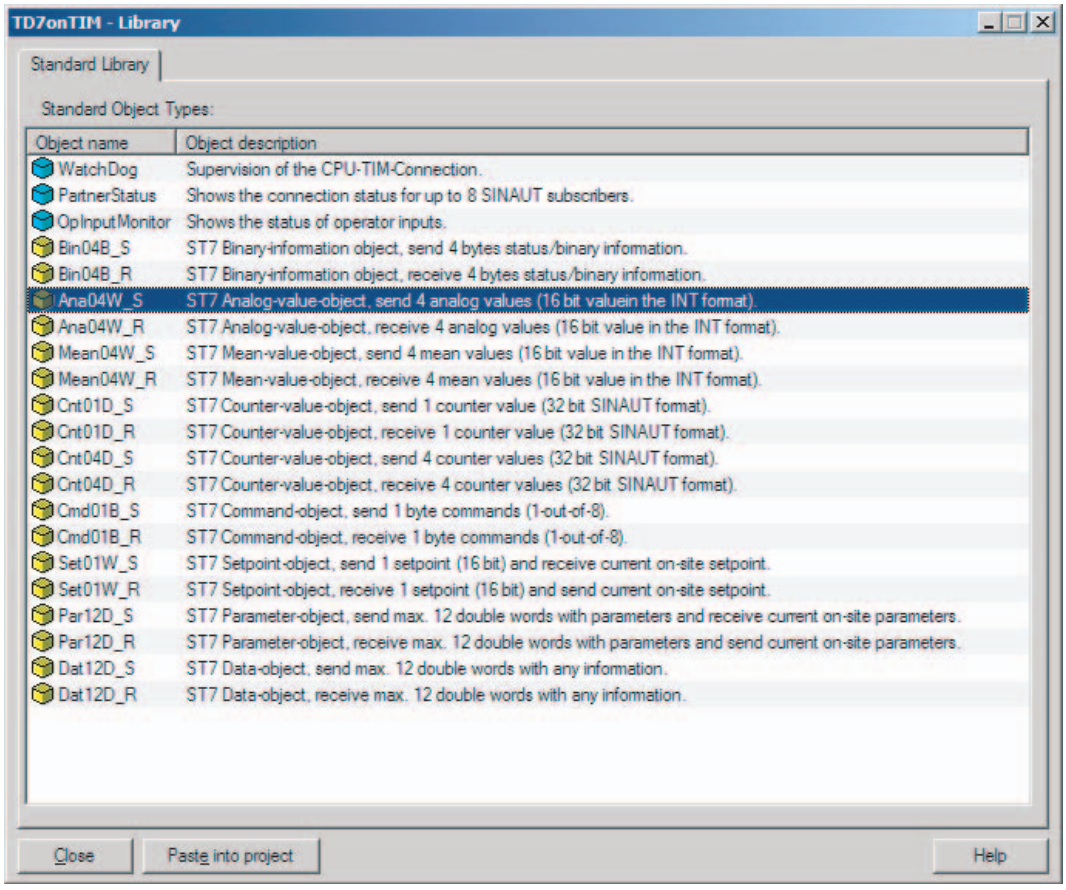

TD7onTIM - Library
Standard Library
Standard Object Types:
Object name
Object description
WatchDog
Supervision of the CPU-TIM-Connection.
PartnerStatus
Shows the connection status for up to 8 SINAUT subscribers.
OpInputMonitor
Shows the status of operator inputs.
Bin04B_S
ST7 Binary-information object, send 4 bytes status/binary information.
Bin04B_R
ST7 Binary-information object, receive 4 bytes status/binary information.
Ana04W_S
ST7 Analog-value-object, send 4 analog values (16 bit valuein the INT format).
Ana04W_R
ST7 Analog-value-object, receive 4 analog values (16 bit value in the INT format).
Mean04W_S
ST7 Mean-value-object, send 4 mean values (16 bit value in the INT format).
Mean04W_R
ST7 Mean-value-object, receive 4 mean values (16 bit value in the INT format).
Cnt01D_S
ST7 Counter-value-object, send 1 counter value (32 bit SINAUT format).
Cnt01D_R
ST7 Counter-value-object, receive 1 counter value (32 bit SINAUT format).
Cnt04D_S
ST7 Counter-value-object, send 4 counter values (32 bit SINAUT format).
Cnt04D_R
ST7 Counter-value-object, receive 4 counter values (32 bit SINAUT format).
Cmd01B_S
ST7 Command-object, send 1 byte commands (1-out-of-8).
Cmd01B_R
ST7 Command-object, receive 1 byte commands (1-out-of-8).
Set01W_S
ST7 Setpoint-object, send 1 setpoint (16 bit) and receive current on-site setpoint.
Set01W_R
ST7 Setpoint-object, receive 1 setpoint (16 bit) and send current on-site setpoint.
Par12D_S
ST7 Parameter-object, send max. 12 double words with parameters and receive current on-site parameters.
Par12D_R
ST7 Parameter-object, receive max. 12 double words with parameters and send current on-site parameters.
Dat12D_S
ST7 Data-object, send max. 12 double words with any information.
Dat12D_R
ST7 Data-object, receive max. 12 double words with any information.
Close
Paste into project
Help

Figure 1-68 Window of the standard library of SINAUT objects for TD7onTIM

Overview of the SINAUT objects

The following data objects are available for TD7onTIM:

Table 1- 5 Overview of the SINAUT objects for TD7onTIM

Object type	Name	Explanation
System objects		
System object	**WatchDog**	Monitoring of the CPU-TIM connection
System object	**PartnerStatus**	Displays connection status for up to 8 SINAUT subscribers
System object	**OpInputMonitor**	Signals detection of hardware entries
Message objects		
Message object for send direction	**Bin04B_S**	Send 4 bytes of messages/binary information
Message object for receive direction	**Bin04B_R**	Receive 4 bytes of messages/binary information
Analog value / mean value objects		
Analog value object for send direction	**Ana04W_S**	Send 4 analog values (16-bit value in the INT format)
Analog value object for receive direction	**Ana04W_R**	Receive 4 analog values (16-bit value in the INT format)
Mean value object for send direction	**Mean04W_S**	Send 4 mean values (16-bit value in the INT format)
Mean value object for receive direction	**Mean04W_R**	Receive 4 mean values (16-bit value in the INT format)
Counted value objects		
Counted value object for send direction	**Cnt01D_S**	Send 1 counted value (32-bit SINAUT format).
Counted value object for receive direction	**Cnt01D_R**	Receive 1 counted value (32-bit SINAUT format)
Counted value object for send direction	**Cnt04D_S**	Send 4 counted values (32-bit SINAUT format)
Counted value object for receive direction	**Cnt04D_R**	Receive 4 counted values (32-bit SINAUT format)
Command objects		
Command object for send direction	**Cmd01B_S**	Send 1 byte commands (1-out-of-8 SINAUT format)
Command object for receive direction	**Cmd01B_R**	Receive 1 byte commands (1-out-of-8 SINAUT format)
Setpoint/parameter objects		
Setpoint object for send direction	**Set01W_S**	Send 1 setpoint (16 bits), object with 3 channels : - operating mode status *local* - returned value - setpoint entry
Setpoint object for receive direction	**Set01W_R**	Receive 1 setpoint (16 bits), object with 3 channels : - operating mode *local* - local setpoint entry - setpoint output

Object type	Name	Explanation
Parameter object for send direction	**Par12D_S**	Send max. 12 double words with parameters or setpoints, object with 3 channels: - operating mode status *local* - returned parameters - parameter entry
Parameter object for receive direction	**Par12D_R**	Receive max. 12 double words with parameters or setpoints, object with 3 channels: - operating mode *local* - local parameter entry - parameter output
Other data objects		
Data object for send direction	**Dat12D_S**	Send max. 12 double words (at least 1 double word) with any information
Data object for receive direction	**Dat12D_R**	Receive max. 12 double words (at least 1 double word) with any information

The endings _S and _R in in the object names mean *S*end or *R*eceive.

The functions of the individual objects are described in detail in the section on setting parameters for the send and receive channels.

Inserting objects in the project

To insert new SINAUT objects in the TD7onTIM of a subscriber, follow the steps outlined below:

1. Go to the directory tree and select the TIM for which you want to configure the SINAUT objects.
2. Open the standard library with the F7 key, the *Standard library* button in the toolbar or using the *SINAUT / Standard Library for the TIM* menu. The library is opened in a separate window. The objects are listed with the object name and a brief object description.
3. In the standard library window, select an object with the mouse.
   To insert several objects at the same time, select an object and press the arrow up or down key while holding the Shift key or select distributed objects one after the other while holding down the Ctrl key. All selected objects are shown on a colored background.
4. In the standard library window, click on the *Paste* button or select *Paste* in the context menu (right mouse button). The selected objects are then added to TD7onTIM. They are all displayed in the list box. Only the data objects are displayed in the directory tree below the selected TIM.
5. You can delete an object you do not require from the TIM directory by selecting it in the directory tree or in the list box and then selecting *Delete*in the context menu (right mouse button).
6. Close the standard library with the *Close* button when you no longer require it.

Note

A maximum of 100 objects can be configured per TD7onTIM.

Copying objects to other TIMs

Once all the SINAUT objects for a TIM have been configured and have had then parameters assigned, you can copy the objects and the parameter assignments to another TIM in the project that requires the same or similar objects.

Follow the steps outlined below:

1. Select the objects in the list box while pressing the Shift key and then press the arrow up or down key or select the objects while pressing the Ctrl key.
2. Select *Copy* in the context menu (right mouse button).
3. Select another TIM in the *TIMs with TD7onTIM* directory.
4. Add the selected objects to this TIM with *Paste* in the context menu.

The objects along with their entire parameter assignment are adopted by the TD7onTIM of the other TIM. In the copied object and its channels, it may be necessary to adapt the subscriber-specific parameter assignment (for example the input and output addresses).

If the complete parameter assignment of TD7onTIM is required for other TIM modules, you can also transfer the entire SINAUT objects to a different TIM. Follow the steps outlined below:

1. Select a TIM in the *TIMs with TD7onTIM* directory.
2. Select *Copy* in the context menu (right mouse button).
3. Select another TIM in the *TIMs with TD7onTIM* directory.
4. Add all objects to the other TIM with *Paste* in the context menu.

1.10.7 Setting parameters for system objects

After the SINAUT objects for the individual subscribers have been configured, you set the parameters for the system objects. Follow the steps outlined below:

1. Select a subscriber in the tree directory.
2. Select the required system object in the list.
   The corresponding parameter assignment dialog is opened below the object list.
3. You set the parameters in this dialog.

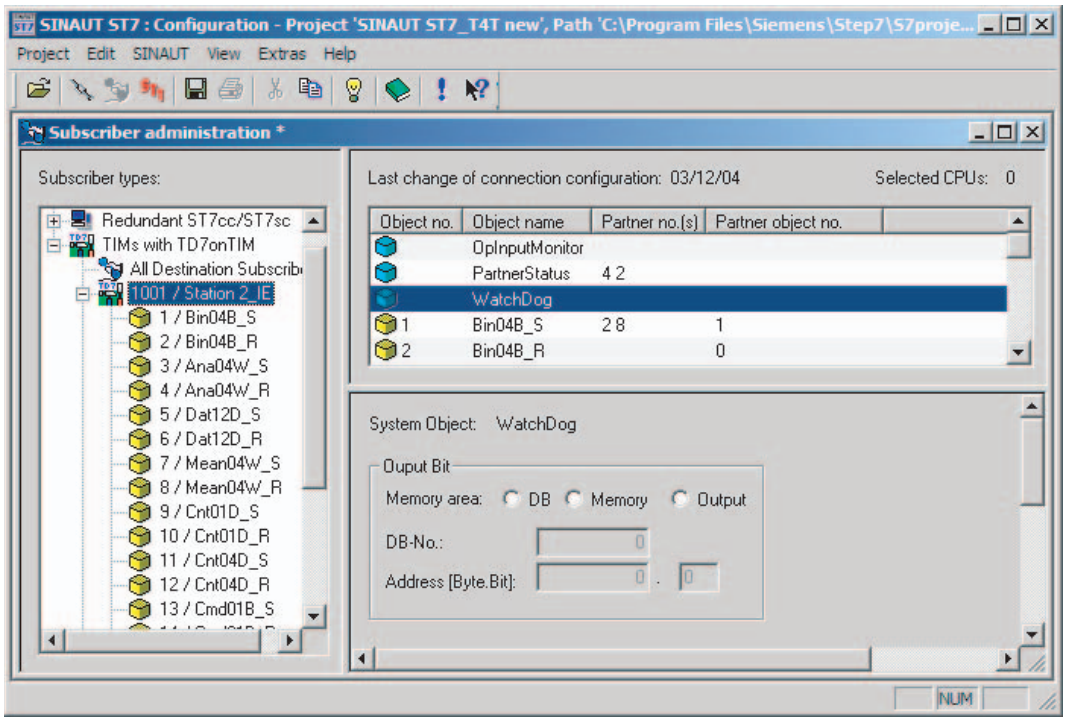


Figure 1-69 TIM with selected system object *WatchDog* and the corresponding parameter assignment dialog

The *WatchDog* system object

The *WatchDog* system object can be included as an option. The WatchDog indicates to the CPU program whether the communication between CPU and local TIM is still working; in other words, whether TD7onTIM is still reading from and writing to the memory areas of the CPU. As long as communication is functioning correctly, a selectable output bit changes state at five second intervals. The constant status change can be evaluated by the CPU user program.

Parameters in the *Output Bit* area

Name:	**Output bit**
Memory area:	The following options are available: - DB: Data block - Memory bit: Memory area - Output: Process output image (PIQ)

DB No.:	Specifies the DB number in the CPU if the data block memory area (*DB*) was selected
Address [Byte.Bit]:	Input fields for the byte and bit number in the selected memory area

The *PartnerStatus* system object

The optional *PartnerStatus* object can be used to monitor the availability of up to 8 communication partners. A partner can be an ST7 CPU or an ST7cc/sc control center to which a connection was configured. TIM modules cannot be monitored with the PartnerStatus object. The status is made available to the CPU user program in an output byte. One bit per communication partner is reserved in the output byte to indicate the status of the respective partner:

- Status 0: Problem on partner or corresponding bit not assigned to any partner
- Status 1: Partner OK

If TD7onTIM has a connection to more than 8 partners whose status needs to be monitored, the *PartnerStatus* system object is configured more than once.

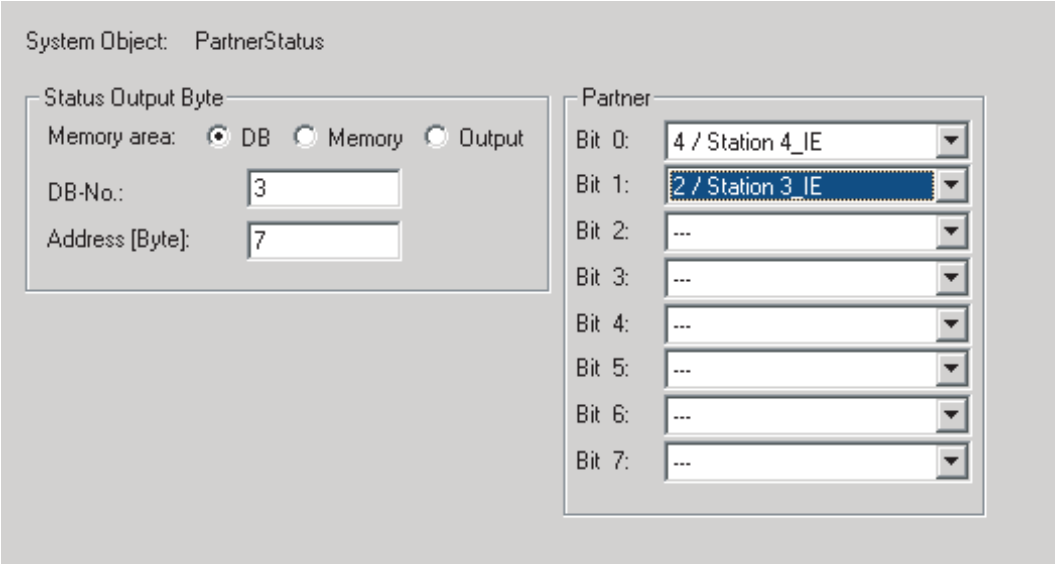


Figure 1-70 Parameter assignment dialog of the *PartnerStatus* system object

Name:	**Status output byte**
Memory area:	The following options are available: - DB: Data block - Memory bit: Memory area - Output: Process output image (PIQ)
DB No.:	Specifies the DB number in the CPU if the data block memory area (*DB*) was selected
Address [Byte]:	Input field for the byte number in the selected memory area

Name:	**Partner**
Explanation:	In the list boxes, the 8 status bits of the output byte are assigned to the communication partners to be monitored. The partners can be selected in the 8 list boxes. The list boxes display only the partners with which the TIM can actually communicate; in other words, to which a connection was configured.
Bit status:	Status = 0: Problem on partner or bit not assigned Status = 1: Partner OK

The *OpInputMonitor* system object

The OpInputMonitor indicates the status of operator input to the CPU (with command, setpoint, and parameter input). The current status can be displayed to the operator for each user program in a suitable form, for example using the LEDs, on an operator panel etc.

Note

The OpInputMonitor system object can only be included once per TD7onTIM.

TD7onTIM should therefore have the OpInputMonitor system object added when one or more of the following objects is used with this TD7onTIM:

- Cmd01B_S (Command object for send direction)
- Set01W_S (Setpoint object for send direction)
- Par12D_S (Parameter object for send direction)

The OpInputMonitor is recommended particularly when commands are entered over digital inputs, for example using buttons connected to them. This also applies to the situation when setpoint and parameter entries are transmitted as the result of the send trigger signal and when this triggering is over a digital input, for example, once again using a button.

Using OpInputMonitor reduces the risk of incorrect input when the entries are made over digital inputs. For these inputs, a *Minimum input time* can be specified for OpInputMonitor, in other words, the button must be pressed for the minimum time. Accidental activation of a button does not then lead to unwanted command, setpoint or parameter transfer. When the minimum input time has elapsed and the button can be released, OpInputMonitor indicates this with its *operator input status byte* in the *Input OK* bit.

Apart from the minimum duration, a *maximum input time* can also be set for digital inputs. This allows a button that is sticking or defective digital inputs that supply a permanent 1 signal to be detected in good time. Such errors are once again indicated in the *operator input status byte* of OpInputMonitor, in this case in the *input error* bit.

The two times and the code bits mentioned above are relevant only for operator input over digital inputs.

For *all* types of operator input, in other words both for input over digital inputs as well as input over the memory or data blocks, OpInputMonitor also returns the error status *1 out-of-n error*. This is set when TD7onTIM has detected one of the following input errors:

- More than 1 bit was set in the input byte of the command object Cmd01B_S. To increase reliability of command input, only one bit may ever be set with this object. If two or more bits are set at the same time, the command input is rejected
- If increased reliability is required for the input of commands, setpoints and parameters, all objects with which this data is sent should be assigned to the fast cycle. All command, setpoint and parameter objects in the fast cycle are subjected to a 1-out-of-n check; in other words, at the end of the fast cycle there is a check to make sure that there is a command, setpoint or parameter entry for only one of the acquired objects. Only then is the corresponding entry processed and transferred. If there is more than one entry, the entries are rejected. A new *command*, *setpoint* or *parameter* is processed only when previously no entry was acquired in at least one fast cycle.

Note

If commands are entered over a memory or data byte, or a setpoint or parameter entry is enabled by a memory or data bit (over the send trigger *Triggersignal*), the set command bit or trigger signal is automatically reset to zero by TD7onTIM. If, however, a 1-out-of-n error is detected, these bits are not automatically reset. They must then be reset by the user or the user program.

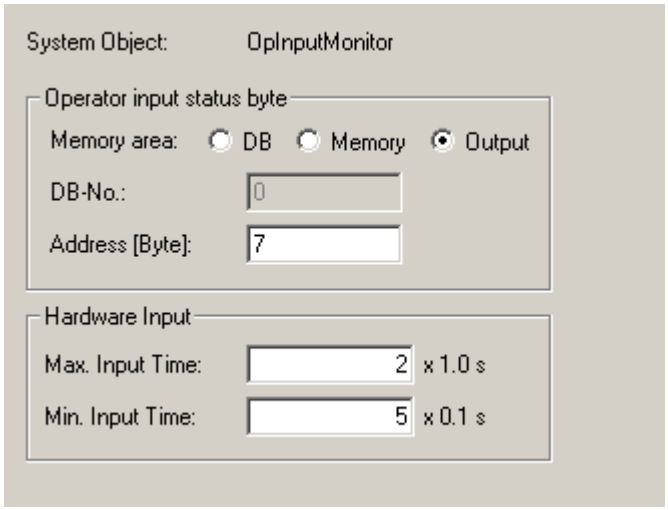


Figure 1-71 Parameter assignment dialog of the *OpInputMonitor* system object

Parameter settings in the *operator input status byte*:

Name:	**Operator input status byte**
Memory area:	The following options are available: - DB: Data block - Memory bit: Memory area - Output: Process output image (PIQ)
DB No.:	Specifies the DB number in the CPU if the data block memory area (*DB*) was selected

Address [Byte]:	Input field for the byte number in the selected memory area
Explanation:	The *operator input status byte* is the output by of the *OpInputMonitor* system object. In the *operator input status byte*, the next 3 bits are assigned (explanation see above).

Byte assignment of the *operator input status byte*								
Bit:	.7	.6	.5	.4	.3	.2	.1	.0
Status:	8	7	6	5	4	1-out-of-n error	Input error	Input OK
For value:	0	0	0	0	0	1	1	1
Unused bits are set to 0								

Parameter settings in the *Hardware input* area:

Name:	**Max. Input Time**
Range of values:	Enter value x 1 [s] (10 corresponds to 10 seconds)
Default:	0
Explanation:	Monitoring time for commands entered over hardware inputs, or setpoints and parameters whose transmission is triggered over a hardware input. If the 1 signal is set at these hardware inputs for longer than defined in *Max. Input Time*, then the *input error* bit is set in the *operator input status byte*. Further hardware entries are not processed as long as the *input error* bit is set. The *Max. Input Time* is specified in seconds. A time of at least 30 seconds is recommended (entry: 30). 0 (zero) can be entered if the parameter is not required.

Name:	**Min. Input Time**
Range of values:	Enter value x 0.1 [s] (10 corresponds to 1 second)
Default:	0
Explanation:	Delay time for commands entered over hardware inputs, or setpoints and parameters whose transmission is triggered over a hardware input. The message is entered in the send buffer of the TIM only if the currently entered command, setpoint for parameter remains unchanged for the specified delay time and no other command or setpoint input is detected during this time. The *Min. Input Time* is specified in tenths of seconds. A time of at least 1 second is recommended (entry: 10). 0 (zero) can be entered if the parameter is not required.

1.10.8 Basic parameters of the data objects

The parameters for data objects are set in two phases:

1. Setting of the basic parameters of the data objects (for example partners to which the data of the object is sent or from which it is received)
2. Setting of the channel-specific properties for the individual send and receive channels

This section first describes setting the basic parameters of the data objects.

Follow the steps outlined below:

1. Select a TIM with TD7onTIM in the directory tree.
2. Select the required data object in the list box. The corresponding parameter assignment dialog opens.
3. You set the parameters in this dialog.

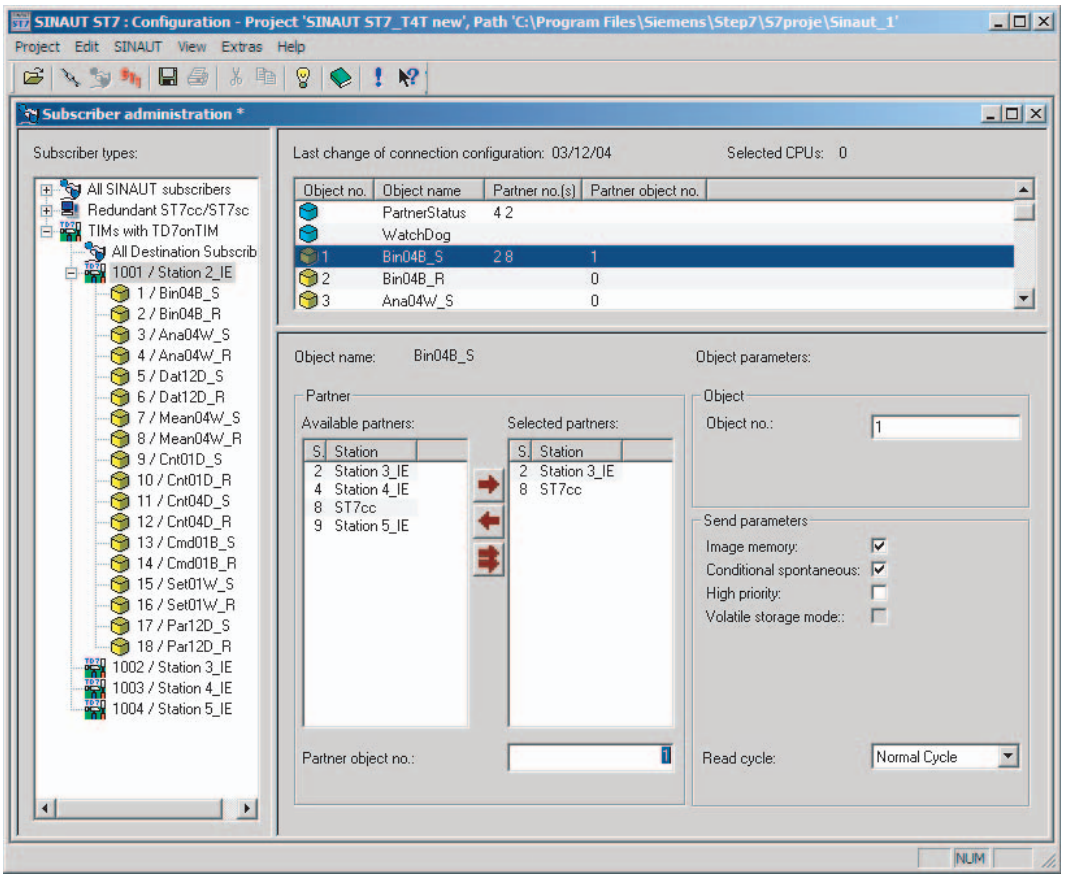


Figure 1-72 Subscriber administration with the parameter assignment dialog of the basic parameters of a data object

Parameters in the *Object* area

Name:	**Object no.**
Range of values:	1 ... 32000
Explanation:	The source object number of this TIM module is set in the input box. The configuration tool proposes a consecutive number that can be modified. An inconsistent duplicate assignment of numbers is prevented.

Parameters in the *Partner* area

The *Available partners* list box shows all the partners configured for the TIM along with their subscriber numbers and station names. The communication partners for the relevant data object are selected from this list and added to the *Selected partners* list.

Note

All partners from which data of the object is received or to which it is sent must be added to the *Selected partners* list. If no partner is entered in the *Selected partners* list, the object is not processed.

Name:	**Selected partners**
Explanation:	The *Selected partners* list displays the communication partners for the relevant data object. These are selected from the *Available partners* list box. Partners are entered in the *Selected partners* list by selecting one or more (using the Ctrl key) partners in the *Available partners* list and clicking on the button with the arrow pointing to the right. The selected partners are then entered in the *Selected partners* list. The button with the double arrow is is used to enter all available partners. Partners are removed from the *Selected partners* list using the button with the arrow pointing left. Up to 15 partners can be configured per data object.

Name:	**Partner object no.**
Range of values:	0 ... 32000

Default:	0
Explanation:	In this input box, you assign the data object to the corresponding partner object of *all* selected communication partners when the partner object number is identical for all partners. Otherwise set 0 (zero). **Note:** There is no object no. for objects of an ST7cc/sc control center! For a send object of TD7onTIM that transfers data to ST7cc/sc, a partner object no. = 0 can be set since the specified partner object no. is not evaluated by ST7cc/sc. For a receive object of TD7onTIM that receives data from ST7cc/sc, the partner object no. = 0 must be set. After object no. = 0 for *send objects*: The partner object no. = 0 must be set if the object data will be sent to several partners and the receiving objects on these partners have different object numbers. If partner object no. = 0 is specified and if *TD7onCPU* is used on the partner, the ListGenerator must be installed on the partner. Only then can the corresponding receive object to be recognized on this partner. Partner object no. = 0 for *receive objects*: The partner object no. = 0 must be set in the object receives data from several partners and the sending objects on these partners have different object numbers. **Note:** If partner object no. is specified for at least one received object, the option *Check of source address* must be deactivated in the basic settings of this TD7onTIM. Otherwise, messages intended for these receive objects will be rejected.

Note

With data objects of the type *Command snd*, *Setpoint send* and *Parameter send*, partner object no. 0 (zero) is not permitted.

The *Send parameters* area:

Name:	**Image memory**
Range of values:	Function active: Send using the image memory principle Function deactivated: Send using the send buffer principle

Default:	In general: Function active With Cmd01B_S, Set01W_S and Par12D_S: Function deactivated and setting cannot be changed
Explanation:	The basic setting for saving messages in the send buffer of the TIM prior to transmission is the *image memory method* set as default in the *Image memory* parameter. As long as a message has not been sent, the process data in the message waiting to be sent is updated by current process data if this changes. This setting is practical for most data. It ensures efficient use of memory for storing messages on the TIM and produces as little message traffic on the WAN link as possible. Generally, the default setting for the *Image memory* parameter only needs to be changed to the *Send buffer method* with a few objects whose data changes must be stored individually on the TIM and sent individually to the partner, for example alarms with a time stamp or analog values with a time stamp for entry in archives. With the *Bin04B_S* object, selected individual binary inputs can be transmitted using the send buffer method even if the *Image memory* function is active. You make this setting during the parameter assignment of the channel type *Binary send* in the *Send buffer principle mask* parameter.

Name:	**Conditional spontaneous**
Range of values:	Function active: Transmission is conditional spontaneous Function deactivated: The transmission is unconditional spontaneous
Default:	Function active
Explanation:	If the function is activated (transmission is conditional spontaneous), the message does not trigger connection establishment in dial-up networks. When using a dedicated lines or Ethernet, this parameter has no significance since transmission is then always immediate. If the function is deactivated (transmission is unconditional spontaneous), the message triggers connection establishment immediately in dial-up networks. In the *Bin04B_S* object, even if the *Conditional spontaneous* function is activated, selected individual binary inputs can trigger *unconditional spontaneous* transmission. You make this setting during the parameter assignment of the channel type *Send binary value* in the *Alarm mask* parameter.

Name:	**High priority**
Range of values:	Function active, function deactivated

Default:	Function deactivated
Explanation:	Important process data can be given higher priority than the messages waiting in the send buffer. If you activate the *High priority* function, the messages of this object are given a higher priority and sent before the other buffered messages. **Note:** In dial-up networks, the *High priority* function does not necessarily lead to immediate connection establishment. This happens only when the *Conditional spontaneous* function has been deactivated for this object.

Volatile storage mode output box:
The *Volatile storage mode* output box indicates that a message already stored in the send buffer of the TIM will be deleted if the subscriber is not available. As long as a subscriber is disrupted, no new non-retentive messages for this subscriber can be entered in the send buffer of the TIM. The *volatile storage mode* applies only to the send messages of the following objects:
- *Cmd01B_S*
- *Set01W_S*
- *Par12D_S*

By deleting command, setpoint and parameter messages, you avoid out-of-date commands, setpoints all parameters being sent to the destination subscriber when a connection is re-established following a failure.

The messages of all other send objects are not deleted when there is a connection failure and further messages can be entered in the send buffer of the TIM during a disruption.

Name:	**Read cycle**
Range of values:	1: *Normal cycle* 2: *Fast cycle*
Default:	Normal cycle
Explanation:	Each data object that sends data must be assigned to one of the two read cycles. The *normal cycle* is the most suitable selection for most process data. Data that must be acquired quickly such as alarms and pulse messages are suitable for assignment to the *fast cycle*. Command, setpoint and parameter objects for which a 1-out-of-n check is required, must be assigned to the *fast cycle*. If these objects are acquired in the normal cycle, they are not included in the 1-out-of-n check. All send channels of a data object are included in the same read cycle. The parameters of the read/write cycle themselves are set in the basic settings of TD7onTIM.

1.10.9 Channel overview and functions of channel parameter assignment

Overview of the channel types

Each data object has a defined number of channels with default parameters. The number of channels and the data type cannot be changed for an individual data object. Apart from the

setpoint and parameter objects, all data objects have one or more channels of the same type.

The channel types differ in the transmission direction relative to the communication partner and fall into the two classes *Send* and *Receive* channels:

- **Send channels** for the *send data* function:
  - Binary send
  - Analog send
  - Data send
  - Mean value send
  - Counted value send
  - Command send
  - Setpoint send
- **Receive channels** for the *receive data* function:
  - Binary receive
  - Analog receive
  - Data receive
  - Mean value receive
  - Counted value receive
  - Command receive
  - Setpoint receive

Dialogs for setting channel parameters

To set the parameters for the channels, you select the data object of a TIM with TD7onTIM in the directory tree. The list box displays the channels of this object with the channel name and channel type. If channels have already had parameters set, the input or output address is displayed in the list.

Below the list box, there is a parameter assignment dialog for the channel selected automatically or with the mouse in the list box.

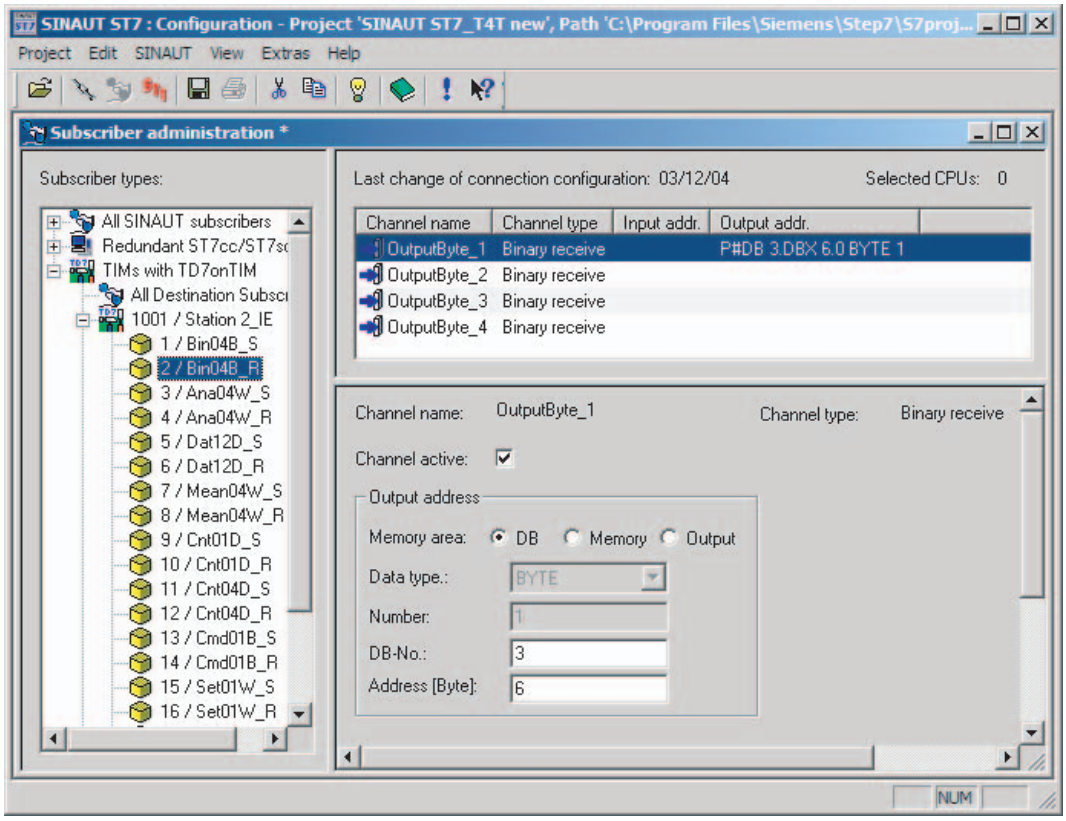


Figure 1-73 Selected object *Bin04B_R* with *Binary receive* channel selected in the list box and its parameter assignment dialog

Activating channels

Prior to parameter assignment, the send and receive channels are not yet active, the input boxes in the dialog are disabled.

Each required channel must be enabled in its parameter assignment dialog using the *Channel active* option.

If certain channels of a data object are not required, they can be ignored.

If users are not sure whether they actually require a channel that has already had parameters set, or when they want to disable a channel later (perhaps temporarily), they can deactivate each individual channel here without losing the parameter settings.

Copying channels

To simplify parameter assignment, channels along with their parameter assignment can be copied. If a data object requires more than one channel with the same parameter assignment (except for the input or output address) and if the parameter settings have been completed for one channel, this channel along with its parameters can be copied.

Follow the steps outlined below:

1. Select a channel in the list box.
2. Select *Copy* in the context menu (right mouse button).

3. Select a different channel of the same object or the channel of another data object of the same type.
4. Insert the channel with its parameters using the *Paste* context menu (right mouse button). The selected channel is overwritten by the channel and its parameters.

Note

When you copy channels, the channel you are copying is not added to the existing channels since the number of channels is fixed per data object. The channel selected prior to paste is overwritten by the copied channel and its parameters.

1.10.10 Mandatory parameters of the send channels

During the parameter assignment of each send channel, you must first set the mandatory parameters *Input address* and *Send trigger*.

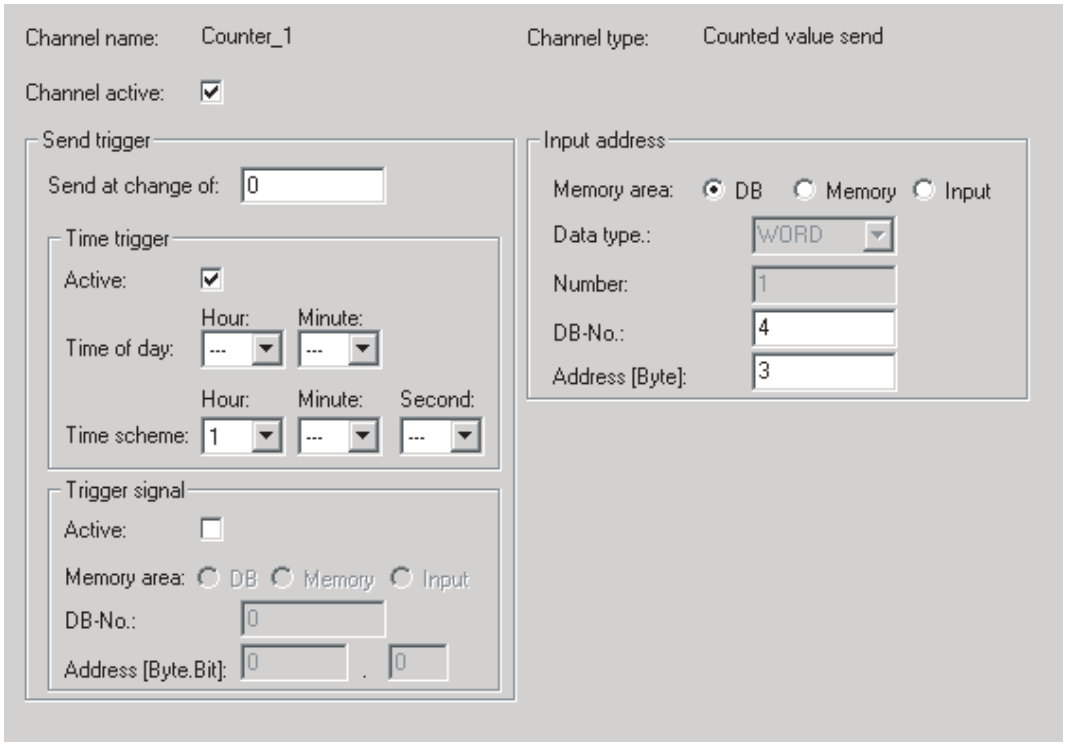


Figure 1-74 Parameter assignment dialog of a send channel based on the example of *Counted value send*

The input address

Data transferred from TD7onTIM to a communication partner is first read from the work memory of the CPU. For each send channel, the source address (*input address*) must be specified in the relevant memory area of work memory on the local CPU module from which the data will be read.

Name:	**Input address**
Memory area:	The following options are available for the source address: - DB: Data block - Memory bit: Memory area - Input: Process input image (PII)
Data type:	The output box displays the default data format of the corresponding channel type: - *Binary send*: BYTE - *Analog send*: WORD - *Data send*: DWORD (double word) - *Mean value send*: WORD - *Counted value send*: WORD - *Command send*: BYTE - *Setpoint send* (object *Set01W_S*): WORD - *Setpoint send* (object *Par12D_S*): DWORD
DB No.:	Specifies the DB number in the CPU if the data block memory area (*DB*) was selected
Address [Byte]:	Input field for the byte number in the selected memory area. For data types involving more than one byte (WORD, DWORD), the least significant byte number must be entered as in STEP 7.
Number:	The number of double words included in the array (maximum 12). The parameter is available only with the following channel types: - *Data send* with the object *Dat12D_S* - *Setpoint send* with object *Par12D_S* The parameter is explained along with these object-specific channels.

Note

Only the specified address areas can be read by TD7onTIM. Data from other areas, for example analog values acquired over peripheral input words (PIW) must be mapped to the bit memory or data block area by the user program.

The send trigger

The second mandatory parameter that must be set for send channels specifies when the data will be sent. This setting is made in the *Send trigger* area.

Four options are available for the send trigger. For each channel, you can configure a single option or a combination of different options.

Note

The send trigger can be set individually for each channel. If an object has more than one channel, remember that the activation of the send trigger of one of the object channels will transmit *all* channels of the data object.

Since the various options of the send trigger must be activated alternatively or additionally with most channel types, the parameter setting should selected to produce practical results.

Name:	**Send at change of**
Range of values:	In general: 0 ... 32767 Channel type *Command send* (object *Cmd01B_S*): 0 or 1 Channel type *Data send* (object *Dat12D_S*): 0 or 1 Channel type *Setpoint send* (object *Set01W_S* or *Par12D_S*) 0 or 1
Defaults:	- for binary value, counted value, data and command channels: 1 - For mean value and setpoint channels: 0 - For analog value channels: 270 (1% of the raw value of S7 analog input modules, 27648 = 100%)
Explanation:	The value must be entered (as an integer) in the input field by which the process value must change so that it is transferred again. If you enter 0 (zero), the function is deactivated.

The *Time trigger* area provides two alternatives *Time of day* or *Time scheme*. A combination of both is not possible.

The *time trigger* send trigger is activated with the *Active* option.

Name:	**Time of day**
Range of values:	Time of day (hour and minute)
Default:	Function not active
Explanation:	If the send trigger is triggered by a time of day, the data is read out at the selected time of day and a message is sent.

Name:	**Time scheme**
Range of values:	Time scheme (hour, minute or second)
Default:	Function not active with the exception of *Mean value send* (*Mean04W_S*): Function active, time scheme: 15 minutes
Explanation:	If the send trigger is triggered by a time scheme, the data is read out at the selected time interval and a message is sent.

If the *time trigger* is, for example, initially set to *time of day* and then to *time scheme*, the previously set times are replaced by dashes (- - -).

The *Trigger signal* area includes the option of an external send trigger that can be activated with the *Active* option.

Name:	**Trigger signal**
Memory area:	The following options are available for the source address: - DB: Data block - Memory bit: Memory area - Input: Process input image (PII)
Default:	Function deactivated
DB No.:	Specifies the DB number in the CPU if the data block memory area (*DB*) was selected
Address [Byte]:	Input field for the byte number in the selected memory area
Explanation:	If a *trigger signal* is specified, the data of the object is transferred when the trigger signal changes from 0 to 1. If the trigger signal is a memory or data bit, it is automatically reset after it has been read. The reset can, if necessary, be evaluated by the user program, for example to display that the message was triggered.

1.10.11 Mandatory parameters of the receive channels

When you set parameters for each receive channel, the *Output address* parameter must be set as a mandatory parameter. This is the address in the memory area of the local CPU to which the received data will be written by TD7onTIM.

As an example of the *Output address* area, we have taken the parameter assignment dialog of the *Binary receive* channel type.

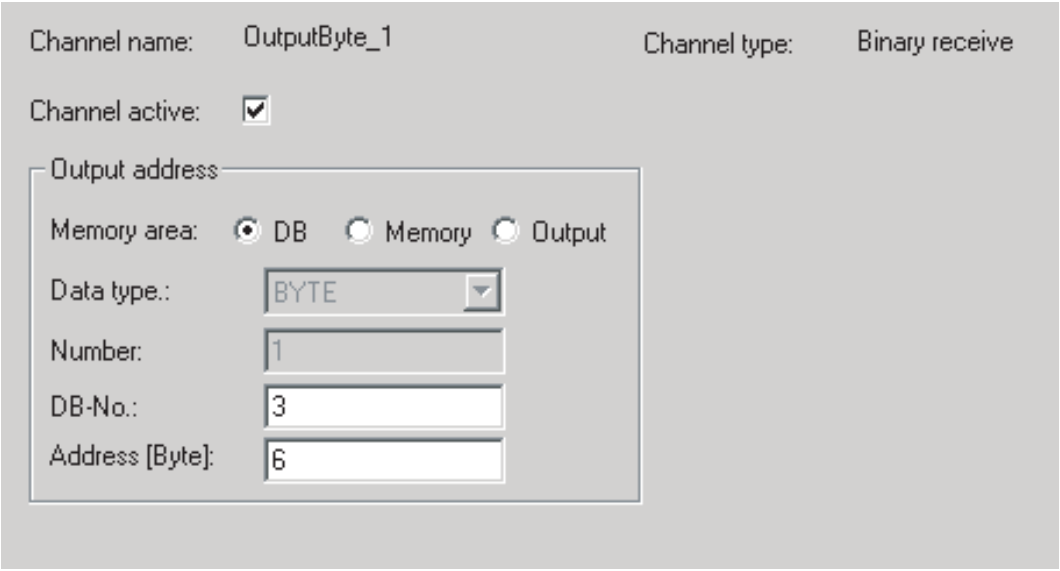


Figure 1-75 Parameter assignment dialog of the *Binary receive* receive channel

The output address

The destination address in the memory area of the CPU is set in the *Output address* area by specifying the following information:

Name:	**Output address**
Memory area:	The following options are available for the destination address: - DB: Data block - Memory bit: Memory area - Output: Process output image (PIQ)
Data type:	The output box displays the default data format of the corresponding channel type: - *Binary receive*: BYTE - *Analog receive*: WORD - *Data receive*: DWORD (double word) - *Mean value receive*: WORD - *Counted value receive*: DWORD - *Command receive*: BYTE - *Setpoint receive* (object *Set01W_R*): WORD - *Setpoint receive* (object *Par12D_R*): DWORD
DB No.:	Specifies the DB number in the CPU if the data block memory area (*DB*) was selected
Address [Byte]:	Input field for the byte number in the selected memory area. For data types involving more than one byte (WORD, DWORD), the least significant byte number must be entered as in STEP 7.
Number:	The number of double words included in the array (maximum 12). The parameter is available only with the following channel types: - *Data receive* with object *Dat12D_R* - *Setpoint receive* with object *Par12D_R* The parameter is explained along with these object-specific channels.

Note

Only the specified address areas can be written by TD7onTIM. Data for other areas, for example analog values output over peripheral output words (PQW) must be mapped over the bit memory or data block area by the user program.

Apart from these generally valid channel parameters, most send and receive channels have specific parameters that are described below based on the individual data objects.

1.10.12 Specific channel parameters of the data objects

Object type *Bin04B_S*

- Channel type: *Binary send*:
  The *Masks* field provides three options for transmitting binary value messages. You can specify bit-by-bit whether certain bits do not trigger message transmission or which bits

trigger a different type of transmission than was specified in the basic parameters of the object. This setting is made in the *Masks* area.

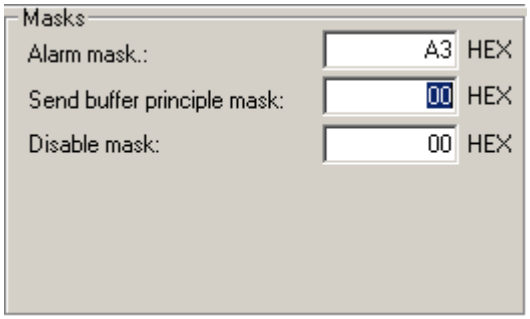


Figure 1-76 The *Masks* area in the parameter assignment dialog of the *Binary send* channel type

Name:	**Alarm mask**
Range of values:	Mask in hexadecimal format
Default:	00 (hex)
Explanation:	Changes in masked bits in the byte of the *Binary send* channel cause an unconditional spontaneous message transmission. Changes from 0 to 1 and from 1 to 0 are evaluated. The corresponding bits are masked in hexadecimal format. The *alarm mask* is only practical when the object is transmitted over a dial-up network and the option *Conditional spontaneous* was activated in the basic parameters of the object.

Name:	**Send buffer principle mask**
Range of values:	Mask in hexadecimal format
Default:	00 (hex)
Explanation:	Changes in masked bits in the byte of the *Binary send* channel cause a message transmission according to the send buffer principle. Changes from 0 to 1 and from 1 to 0 are evaluated. The corresponding bits are masked in hexadecimal format. The*Send buffer principle mask* is only practical when the option *Image memory* was activated in the basic parameters of the object.

Name:	**Disable mask**
Range of values:	Mask in hexadecimal format

Default: 00 (hex)

Explanation: Masked bits in the byte of the *Binary send* channel are ignored when changes are checked. This means that changes to the masked bits for this channel do not trigger message transmission. A masked bit always has the value 0 in the message.

The corresponding bits are masked in hexadecimal format.

The bits are masked as shown in the following example in which the hexadecimal value *A3* is entered in the input field of the parameter assignment dialog. The bits with the value 1 are masked; in other words bits no. 0, 1, 5 and 7 cause the relevant function in the described masks.

Byte assignment								
Bit:	.7	.6	.5	.4	.3	.2	.1	.0
Masked	1	0	1	0	0	0	1	1
Hex mask	A				3			

Object type *Bin04B_R*

- Channel type: *Binary receive*:
  This channel type has no specific parameters.

Object type *Ana04W_S*

- Channel type: *Analog send*:
  The following parameters are available in the *Processing parameters* area:

Name: **Unipolar analog value**

Range of values: Function active, function deactivated

Default: Function active

Explanation: If the function is activated, negative analog values are corrected to the value zero.

The error ID 8000h (-32768), that is displayed, for example, if there is a wire break in *life zero* inputs, is nevertheless transmitted.

Name: **Smoothing factor**

Range of values: 1 = none, 4 = weak, 32 = medium, 64 = strong

Default: 1

Explanation: Using the smoothing factor, quickly fluctuating analog values can be smoothed to a greater or lesser extent depending on the parameter setting. It may then be possible to set a lower value for the send trigger *Send at change of*.

The smoothing factors are identical to the smoothing factors that are configured for some S7 analog input modules. The smoothing in the channel functions according to the same formula as on an input module:

$$y_n = \frac{x_n + (k-1)y_{n-1}}{k}$$ where

y_n = smoothed value in the current cycle n

y_n = acquired value in the current cycle n

k = smoothing factor

Name: **Fault suppression time**

Range of values: 0 ... 32767 [s]

Default: 0

Explanation: Transmission of an analog value located in the overflow or underflow range (7FFFh or 8000h) is suppressed for the time period specified here. The value 7FFFh or 8000h is only sent after this time has elapsed, if it is still pending. If the value returns to below 7FFFh or above 8000h again before this time elapses, it is immediately sent again as normal. The suppression time is started again for the full duration the next time 7FFFh or 8000h is received.

This is typically used for temporary suppression of current values that may occur when powerful motors are started. The analog input may exceed several times the maximum range under some circumstances. Suppression prevents these values from being signaled as faults in the control center system.

The suppression is adjusted to analog values that are acquired by the S7 analog input modules as raw values. These modules return the specified values for the overflow or underflow range for all input ranges (also for life-zero inputs).

When the user provides specific values, fault suppression is only possible if these also adopt the values 7FFFh or 8000h when the permitted ranges are exceeded. If this is not the case, the parameter does not need to have a value entered.

Object type *Ana04W_R*

- Channel type: *Analog receive*:
  This channel type has no specific parameters.

Object type *Dat12D_S*

- Channel type: *Data send*:
  With the channel type *Data send*, a data field of a maximum of 12 double words can be sent. This setting is made with the *Number* parameter in the *Input address* area.

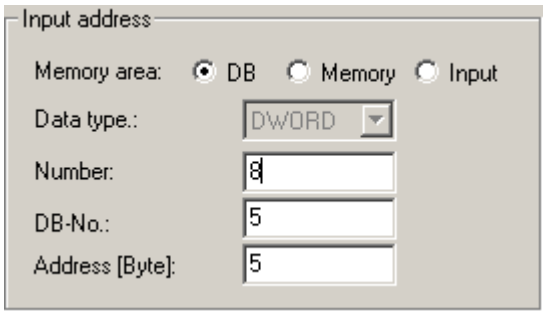


Figure 1-77 Section of the parameter assignment dialog of the channel type *Data send* with the *Number* parameter

Name:	**Number**
Range of values:	1 ... 12
Default:	12
Explanation:	The parameter decides the size of the data field in double words. This allows the message length to be reduced to the length actually required. This saves transmission time.

Object type *Dat12D_R*

- Channel type: *Data receive*:
  With the channel type *Data receive*, a data field of a maximum of 12 double words can be received. This setting is made with the *Number* parameter in the *Output address* area.

Name:	**Number**
Range of values:	1 ... 12
Default:	12
Explanation:	The parameter specifies the size of the data field in double words that must be identical to the data field size of the sending partner object *Dat12D_S*.

Object type *Mean04W_S*

- Channel type: *Mean value receive*:
  This channel type has no specific parameters. The duration of the interval for forming an individual mean value is determined by the *Time trigger* specified for the *Mean value send* channel.
  If the mean value is to be entered in an archive in the control center, the *Mean04W_S* object should be transmitted according to the send buffer principle.

Object type *Mean04W_R*

- Channel type: *Mean value receive*:
  This channel type has no specific parameters.

Object types *Cnt01D_S* and *Cnt04D_S*

The object type *Cnt01D_S* sends a counted value, *Cnt04D_S* sends four counted values.

- Channel type: *Counted value send*:
  This channel type has no specific parameters.
  The counted value read in from the CPU must be made available by a software counter of the CPU in the WORD format.
  In TD7onTIM, the value originating from the CPU counter is compared with the value last read from the object and the difference is added to the SINAUT counted value maintained internally in the *Cnt01D_S* or *Cnt04D_S* object. An overflow of the CPU counter at 65535 is detected by the counted value object and taken into account.
  The internally formed SINAUT counted value is stored by TD7onTIM in DWORD format and transmitted.

Object types *Cnt01D_R* and *Cnt04D_R*

The object type *Cnt01D_R* receives a counted value, *Cnt04D_R* receives four counted values.

- Channel type: *Counted value receive*:
  The SINAUT counted value received from the partner object is compared with the last received counted value and the difference is added to the value at *CountedValueOutput*. The value is output in DWORD format so that the maximum displayable counted value is 2,147,483,647. If the maximum value that can be represented is exceeded, the counted value starts again at 0 and counting continues in the positive numeric range.

  Reset:
  If the counted value at *CountedValueOutput* is reset to zero, when the next counted value is received, the difference is added to zero. To reset, a bit in a memory area of the CPU is defined in the *Reset* field. The reset takes place on a signal edge change from 0 to 1.

Name:	**Reset**
Memory area:	The following options are available for the address: - DB: Data block - Memory bit: Memory area - Input: Process input image (PII)
Default:	Function inactive
DB No.:	Specifies the DB number in the CPU if the data block memory area (*DB*) was selected
Address [Byte.Bit]:	Input fields for the byte and bit number in the selected memory area

Object type *Cmd01B_S*

If increased reliability is required for the input of commands, setpoints and parameters, all objects with which this data is sent should be assigned to the fast cycle. All command, setpoint and parameter objects in the fast cycle are subjected to a 1-out-of-n check; in other

words, at the end of the fast cycle there is a check to make sure that there is a command, setpoint or parameter entry for only *one* of the acquired objects. Only then is the corresponding entry processed and transferred. If there is more than one entry, the entries are rejected. A new command, setpoint or parameter is processed only when previously no entry was acquired in one fast cycle. The error status is indicated in the output byte of the *OpInputMonitor* system object using the 1-out-of-n error bit.

- Channel type: *Command send*:
  Only one input may be set at the same time for command input. This is verified by the 1-out-of-8 check. If a 1-out-of-8 error is detected, the entered commands are not processed further. A new command is only processed again when there was previously no command pending for one read cycle. The error status is displayed in the output byte of the *OpInputMonitor* system object in bit 2 (*1-out-of-n error*). The bit remains set until the error is corrected.

  If the command is entered over *digital inputs*, for example using a button connected to them, the button must remain pressed until it is acquired by TD7onTIM. The output byte of the OpInputMonitor system object indicates when the command has been acquired in the Input OK bit. OpInputMonitor also takes into account any minimum input time that has been set for it; in other words the button must be pressed at least as long as this selected time. Only then is *Input OK* indicated.
  If the commands are input over *memory* or *data* bits, the set bit is automatically reset by TD7onTIM as soon as it is acquired. Here, there is *noInput OK* display. Successful acquisition can, however, be recognized indirectly because the command bit was reset.

  Special feature of send trigger:
  For the *Send at change of* send trigger, only the values 0 and 1 can be set. Values higher than 1 are meaningless for command input.

  Disable mask:
  For the *Command send* channel type, individual bits can be masked for command acquisition. You do this in the *Disable mask*:

Name:	**Disable mask**
Range of values:	Mask in hexadecimal format
Default:	00 (hex)
Explanation:	Masked bits in the byte of the *Command send* channel are ignored when changes are checked. This means that changes to the masked bits for this channel do not trigger message transmission. A masked bit always has the value 0 in the message. The corresponding bits are masked in hexadecimal format.

The bits are masked as shown in the following example in which the hexadecimal value *A3* is entered in the input field of the parameter assignment dialog. The bits with the value 1 are masked; in other words bits no. 0, 1, 5 and 7 are ignored in command acquisition.

Byte assignment								
Bit:	.7	.6	.5	.4	.3	.2	.1	.0
Masked	1	0	1	0	0	0	1	1
Hex mask	A				3			

Object type *Cmd01B_R*

- Channel type: *Command receive*:

Name:	**Command output time**
Range of values:	0 ... 50 [s x 0.1] (5 = 0.5 seconds)
Default:	5
Explanation:	This is the command output time for the command outputs of the channel. When the set time has elapsed, the command output is reset again by TD7onTIM. The *command output time* applies to all 8 command outputs of the *Command receive* channel. If the *command output time* is set to 0, a set command output is not reset by TD7onTIM. Resetting the command output to zero must then be implemented in the user program.

Object type *Set01W_S*, the setpoint object for the send direction

The object type *Set01W_S* sends 1 setpoint. The setpoint assignment status 'local' and the locally valid setpoint can also be indicated by this object.

If increased reliability is required for the input of commands, setpoints and parameters, all objects with which this data is sent should be assigned to the fast cycle. All command, setpoint and parameter objects in the fast cycle are subjected to a 1-out-of-n check; in other words, at the end of the fast cycle there is a check to make sure that there is a command, setpoint or parameter entry for only *one* of the acquired objects. Only then is the corresponding entry processed and transferred. If there is more than one entry, the entries are rejected. A new command, setpoint or parameter is processed only when previously no entries were acquired in one fast cycle. The error status is indicated in the output byte of the OpInputMonitor system object using the 1-out-of-n error bit.

If a setpoint entry is transmitted as a result of the *Trigger signal* send trigger, and if this is triggered over a digital input, for example by a connected button, the button must remain activated until the signal has been acquired by TD7onTIM. The *operator input status byte* of the *OpInputMonitor* system object indicates whether the entry has been acquired using the *Input OK* bit.
OpInputMonitor takes into account any minimum input time; in other words, the button must be kept pressed for at least as long as the time set with this parameter. Only then is *Input OK* indicated.
If the trigger signal is a *memory* or *data* bit, the bit is automatically reset by TD7onTIM as soon as it is acquired. Successful acquisition can be recognized indirectly because the trigger bit was reset.

- Channel name: *LocalOperation* - Setpoint assignment status 'local'
  (Channel type: *Binary receive*):
  The return message from the partner object that the local object is set to 'local operation' is sent over this channel. The *LocalOperation* channel is used only for signaling. It can but does not necessarily need to be used.
  A setpoint can also be set locally at the partner object that receives the setpoint. As information, the input parameter *Local* can then be set to 'local' locally on the partner object. The current status of the *Local* input parameter is reported by the partner object and displayed here at the *LocalOperation* output.
  An interlock of the remote and local setpoint assignment must be implemented by the user program; in contrast to TD7onCPU, in TD7onTIM, this parameter (or the *Local*

parameter of the partner object *Set01W_R*) does not cause an interlock with local setpoint assignment.
After the local or the partner CPU has started up or after the return of the connection, a general request makes sure that the current valid status of the partner is indicated in *LocalOperation*.

If you do not require the parameter, leave it inactive.
This channel type has no specific parameters.

- Channel name: *ReturnedSetpoint* - returned setpoint
(channel type: *Data receive*):
In TD7onTIM, the partner object receiving the setpoint reports back the currently valid local setpoint when the *LocalSetpointInput* channel is set there. This returned value is displayed at the *ReturnedSetpoint* output. If the partner object is set to 'local' and if a new entry is made there, the setpoint changed there is indicated at *ReturnedSetpoint* if the *Local* channel is set on the partner object.
After the local or partner CPU has started up or after return of the connection, an automatic general request makes sure that the currently valid local setpoint is indicated at *ReturnedSetpoint*.

  If you do not require the parameter, leave it inactive.
  This channel type has no specific parameters.

- Channel name: *SetpointInput* - Setpoint input
(channel type: *Setpoint send*):
The setpoint to be sent to the partner object is entered over this channel.

  This channel type has no specific parameters.

Object type *Set01W_R*, the setpoint object for the receive direction

The object type *Set01W_R* receives 1 setpoint. The object also has an input over which the locally valid setpoint can be returned.

- Channel name: *Local* - Setpoint assignment 'local'
(Channel type: *Message send*):
The information as to whether the locally set setpoint or the remote setpoint is valid can be signaled over this input. The current status of the *Local* input and a copy of the current local setpoint is returned (mirrored) to the partner. A setpoint sent by the remote partner (for example, master station) can also be accepted when *Local* is set.
In contrast to TD7onCPU, the *Local* cannel here is only for information. An interlock with the remote setpoint assignment must be implemented in the user program.

  If you do not require the parameter, leave it inactive.
  This channel type has no specific parameters.

- Channel name: *LocalSetpointInput* - Local setpoint input
(channel type: *Data send*):
A locally active setpoint can be returned over the *LocalSetpointInput* input to the send block.

  If you do not require the parameter, leave it inactive.
  This channel type has no specific parameters.

- Channel name: *SetpointOutput* - Setpoint output
  (channel type: *Setpoint receive*):
  The setpoint sent by the partner object or entered locally at *LocalSetpointInput* is output at *SetpointOutput*.

  This channel type has no specific parameters.

Object type *Par12D_S*, the parameter object for the send direction

The object type *Par12D_S* sends 1 to 12 parameters. The setpoint assignment status 'local' and the locally valid parameters can also be indicated by this object.

If increased reliability is required for the input of commands, setpoints and parameters, all objects with which this data is sent should be assigned to the fast cycle. All command, setpoint and parameter objects in the fast cycle are subjected to a 1-out-of-n check; in other words, at the end of the fast cycle there is a check to make sure that there is a command, setpoint or parameter entry for only *one* of the acquired objects. Only then is the corresponding entry processed and transferred. If there is more than one entry, the entries are rejected. A new command, setpoint or parameter is processed only when previously no entry was acquired in one fast cycle. The error status is indicated in the output byte of the *OpInputMonitor* system object using the *1-out-of-n error* bit.

If a parameter entry is transmitted as a result of the *Trigger signal* send trigger, and if this is triggered over a digital input, for example by a connected button, the button must remain activated until the signal was acquired by TD7onTIM. The *operator input status byte* of the *OpInputMonitor* system object indicates whether acquisition was successful in the *Input OK* bit. OpInputMonitor also takes into account any minimum input time that has been set for it; in other words the button must be pressed at least as long as this selected time. Only then is *Input OK* indicated.
If the trigger signal is a *memory* or *data* bit, the bit is automatically reset by TD7onTIM as soon as it is acquired. Successful acquisition can be recognized indirectly because the trigger bit was reset.

- Channel name: *LocalOperation* - Setpoint assignment status 'local'
  (Channel type: *Binary receive*):
  The return message from the partner object that the local object is set to 'local operation' is sent over this channel. The LocalOperation channel is used only for signaling. It can but does not necessarily need to be used.
  A parameter can also be entered locally at the partner object that receives the parameter. As information, the input channel *Local* can then be set to 'local' locally on the partner object. The current status of the *Local* input channel is reported by the partner object and displayed here at the *LocalOperation* output.
  An interlock of the remote and local setpoint assignment must be implemented by the user program; in contrast to TD7onCPU, in TD7onTIM, this parameter (or the *Local* parameter of the partner object *Par12D_R*) does not cause an interlock with local setpoint assignment.
  After the local or the partner CPU has started up or after the return of the connection, a general request makes sure that the current valid status of the partner is indicated in *LocalOperation*.

  If you do not require the parameter, leave it inactive.
  This channel type has no specific parameters.

- Channel name: *ReturnedParameter* - Returned parameters
  (Channel type: *Data receive*):
  In TD7onTIM, the partner object receiving the parameter values reports back the

currently valid local parameter values when the *LocalParameterInput* channel is set there. These returned values are displayed at the *ReturnedParameter* output. If the partner object is set to 'local' and if a new entry is made there, the parameters changed there are indicated at *ReturnedParameter* if the *Local* channel is set on the partner object.
After the local or partner CPU has started up or after return of the connection, an automatic general request makes sure that the currently valid local parameters are indicated at *ReturnedParameter*.

If you do not require the parameter, leave it inactive.
This channel type has no specific parameters.

- Channel name: *ParameterInput* - Parameter input
(Channel type: *Setpoint send*):
The parameters to be sent to the partner object are entered over this channel. The content per double word can be a value in double word format (DWORD), however, a mixture of other formats is permitted if they in turn result in a double word, for example
- 4 bytes or
- 2 words or
- 2 bytes plus 1 word.

  The number of required double words, in other words the size of the data field is set in the *Number* input box.

Name:	**Number**
Range of values:	1 ... 12
Default:	12
Explanation:	The parameter decides the size of the data field in double words. This allows the message length to be reduced to the length actually required. This saves transmission time.

Object type *Par12D_R*, the parameter object for the receive direction

The *Par12D_R* object type receives 1 to 12 parameters, for example setpoints. The object also has an input channel over which the locally valid parameters can be returned.

- Channel name: *Local* - Parameter assignment 'local'
(Channel type: *Binary value send*):
The information as to whether the locally set parameters or the remote parameters are valid can be signaled over this input. The current status of the *Local* input and a copy of the local parameters is returned (mirrored) to the partner. A parameter record send by the remote partner (for example, master station) can also be accepted when *Local* is set.
In contrast to TD7onCPU, the channel here is only for information. An interlock with the remote setpoint assignment must be implemented in the user program.

  If you do not require the parameter, leave it inactive.
This channel type has no specific parameters.

- Channel name: *LocalParameterInput* - Local parameter input
  (Channel type: *Data send*):
  Local active parameters can be returned to the send block over the *LocalParameterInput* input.
  The data area corresponds to the data area set for *ParameterOutput* of the same object.

  If you do not require the parameter, leave it inactive.
  This channel type has no specific parameters.

- Channel name: *ParameterOutput* - Parameter output
  (Channel type: *Setpoint receive*):
  The parameters sent by the partner object or enter the local at *LocalParameterInput* are output at *ParameterOutput*.
  The data area can vary length between 1 and 12 double words. The content per double words can be a value in double word format, however, a mixture of other formats his permitted if they in turn result in a double word, for example
  - 4 bytes or
  - 2 words or
  - 2 bytes plus 1 word.

Name:	**Number**
Range of values:	1 ... 12
Default:	12
Explanation:	The parameter decides the size of the *ParameterOutput* data field in double words. This must be identical to the data field size of the sending partner object *Par12D_S*.

1.10.13 Synchronization of the CPU time with TD7onTIM

Time information provided by the TIM

The synchronization of the time on stations is selected in HW Config in the properties dialog of the TIM, *Time Service* tab, refer to the section Setting TIM module parameters (Page 24).

If the "TD7onCPU" TD7 software is running on an Ethernet TIM (TIM 3V-IE variant or TIM 4R-IE) and the TIM's time is synchronized by a master computer, the TIM acts as time master and synchronizes its local CPU (using FC TimeTask) over the backplane bus.

If the "TD7onTIM" TD7 software is running on an Ethernet TIM and the time on the TIM is synchronized by a master computer, the TIM makes the time available to its local CPU at the I/O addresses. The CPU user program can then read and evaluate the time there.

The I/O addresses of the CPU available for the date and time information from the TIM are set in HW Config in the properties dialog of the TIM, *Addresses* tab, refer to the section "Setting TIM module parameters (Page 24)".

The time information of a time-synchronized TIM module with TD7onTIM is stored in 8 bytes of the peripheral "inputs". The time information has an offset of 8 bytes from the base value of the inputs. The following table shows the assignment of the 8 bytes of time information.

Time-of-day format

Table 1- 6 Format of the time information in the inputs of the I/O addresses of the TIM

Byte No.	Offset [bytes]	Meaning	High nibble		Low nibble	
			Value	Range of values (decimal)	Value	Range of values (decimal)
1	+8	**Year**	tens	0...9	ones	0...9
2	+9	**Month**	tens	0...9	ones	0...9
3	+10	**Tag**	tens	0...9	ones	0...9
4	+11	**Hour**	tens	0...9	ones	0...9
5	+12	**Minute**	tens	0...9	ones	0...9
6	+13	**Second**	tens	0...9	ones	0...9
7	+14	**Millisecond**	hundreds	0...9	tens	0...9
8	+15	**Millisecond**	ones	0...9	**Status**	**bit coded**

Meaning of the entries in the table:

- Offset: Offset to the base value of the peripheral inputs in bytes
- High nibble: Bits 4 -7:
- Low nibble: Bits 0-3
- Value: Position of the relevant number
  - Year, month, day, hour, minute and second are two-digit (tens + ones)
  - Milliseconds are three-digit (hundreds + tens + ones)
- Status: Status of the time information

Format of "Status"

The status of the time information is available in the four bits of the low nibble of byte 8. The following table shows the meaning and the values of the status.

Table 1- 7 Status bits of the time (low nibble of byte no. 8)

Bit No.	**3**	**2**	**1**	**0**
Meaning	Prewarning bit	*Not defined*	Daylight saving time (DS), standard time (ST)	Validity of the time
Value	0 = - 1 = prewarning: Changeover at the next full hour (DS -> ST or ST -> DS)		0 = ST 1 = DS	0 = invalid 1 = time valid

Initial setting of the time

The user program that reads out the time on the CPU should only do this when the validity bit is set. This is the case as soon as the TIM is synchronized the first time, either by the time master in the SINAUT network or by the PG.

Note

If the time on the TIM was set from a PG, this is always indicated as standard time (status bit 1 has the value "0").

1.11 Saving and generating system data

1.11.1 Saving subscriber data

You save the data from subscriber administration using the *Save* button in the toolbar or with the *SINAUT / Save* menu. The *Options* dialog opens in which you specify the scope of the system data to be saved, process and prepared.

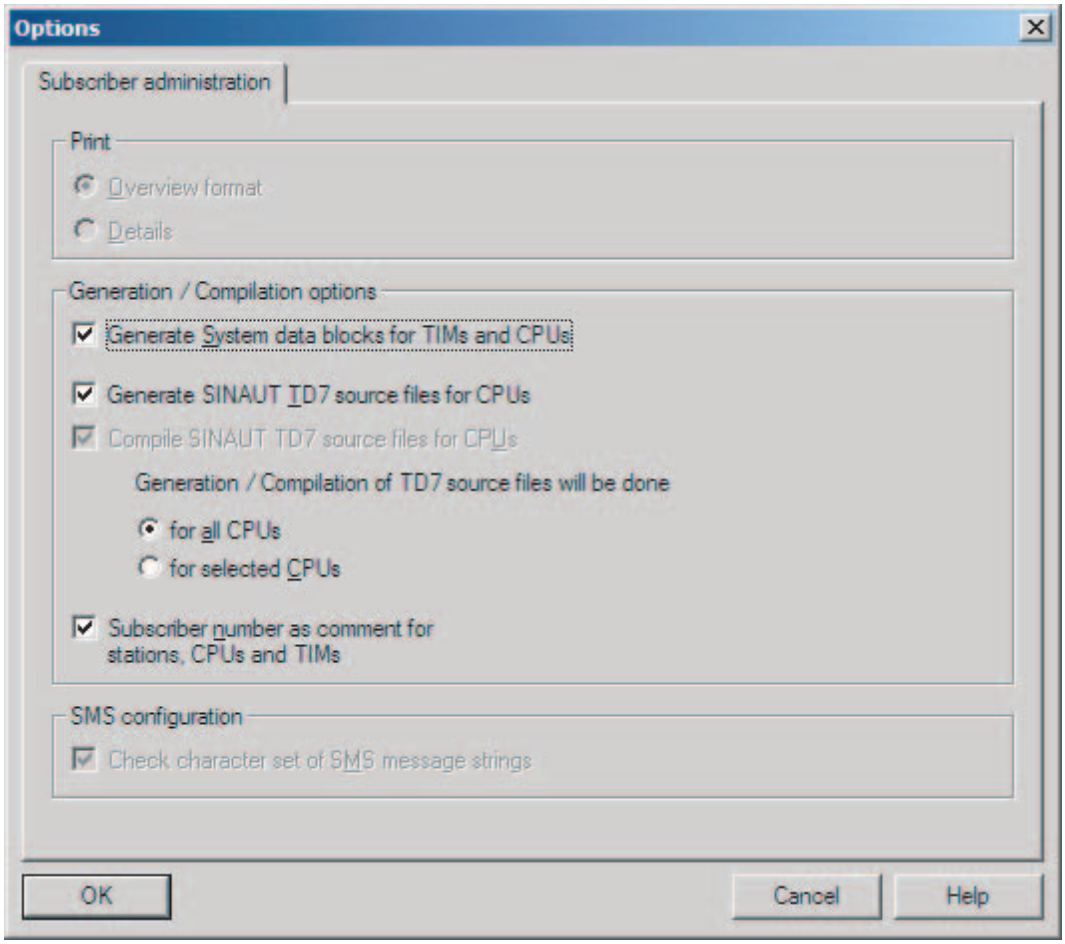


Figure 1-78 The *Options* dialog after selecting the Save function

The *Options* dialog provides the following convenient functions in the *Generation / Compilation options* area:

- Generate System data blocks for TIMs and CPUs
- Generate SINAUT TD7 source files for CPUs

The other options relate to the scope of system data generation

- for all CPUs or
- for selected CPUs (selected in the subscriber list)

The third option relates to generating

- Subscriber number as comment for stations, CPUs and TIMs

The convenient functions described below represent the automation of several steps in the STEP 7 system. These functions always relate to all SINAUT networked subscribers; in other words, to subscribers involved in at least one SINAUT connection.

Regardless of the selected generation options, the internal data is saved and a consistent version is always available later.

After saving the internal data, a consistency check determines whether the user data is free of errors. If this is the case, the required functions are executed.

If problems are detected during the consistency check, the functions are *not* executed. An error list is displayed as well as a message indicating the functions that have not been executed.

As soon as the generation has been completed successfully, this is indicated by a status dialog.

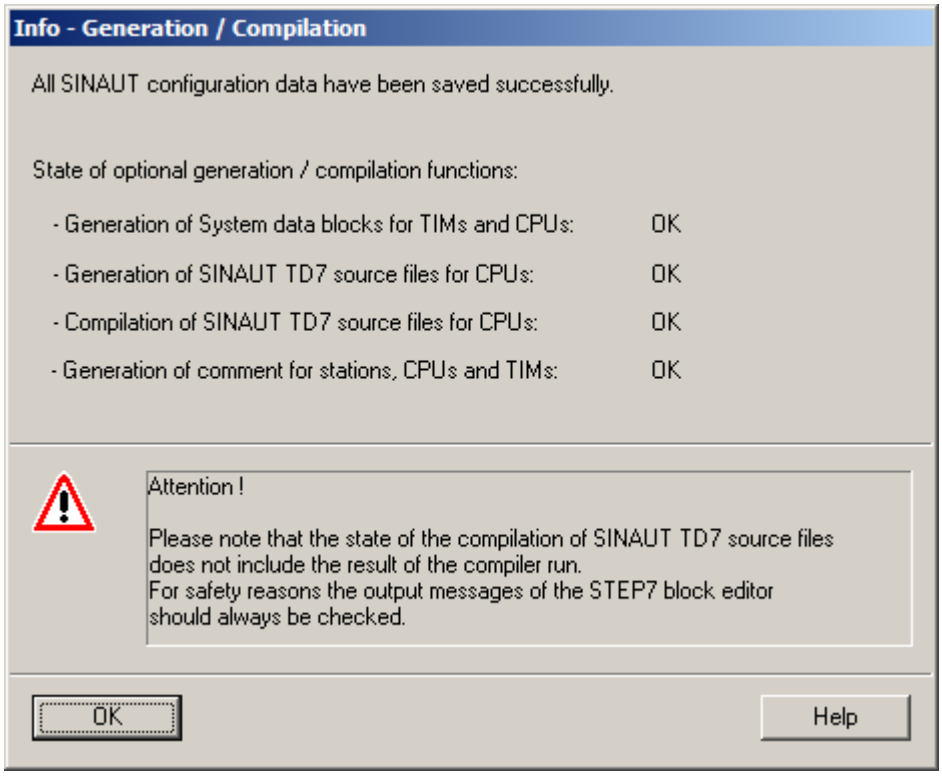

Info - Generation / Compilation
All SINAUT configuration data have been saved successfully.
State of optional generation / compilation functions:
- Generation of System data blocks for TIMs and CPUs: OK
- Generation of SINAUT TD7 source files for CPUs: OK
- Compilation of SINAUT TD7 source files for CPUs: OK
- Generation of comment for stations, CPUs and TIMs: OK
Attention !
Please note that the state of the compilation of SINAUT TD7 source files does not include the result of the compiler run.
For safety reasons the output messages of the STEP7 block editor should always be checked.
OK
Help

Figure 1-79 Status dialog after saving and generating the system data

1.11.2 Generating system data blocks

All the parameters of the TIM module from the hardware setting to information on communication partners or local connections are packed in system data blocks (SDBs). SDBs with numbers starting at 1000 are used.

If S7-homogeneous connections (communication block connections) are used for communication between TIM and TIM or between TIM and CPU, their data is packed in SDBs starting at no. 700.

If the *Generate System data blocks for TIMS and CPUs* option is selected in the *Options* dialog, this system data is saved for all the SINAUT-networked TIM and CPU modules in the subscriber list and saved in the offline data management.

Note

The SDBs must be transferred to the modules either in the SIMATIC Manager or using the SINAUT Diagnostics and Service tool.

1.11.3 Compiling SINAUT TD7 blocks for the CPU

The following descriptions of the TD7 software relate only to the TD7 software for the CPU.

If the *Compile SINAUT TD7 source files for CPUs* option is selected in the *Options* dialog, several compile functions will be activated for all SINAUT-networked CPU modules or only for the CPU modules selected in the subscriber list. These compilation functions include:

- Entry of dynamic data for the SINAUT TD7 software in the form of data blocks in the source file and generation of symbolic entries for the communication DBs.
- Entry of SINAUT TD7 basic blocks that do not exist in the source file or of blocks that exist in a version earlier than or equal to the library version.
- Entry of CPU-specific SINAUT TD7 blocks in the source file according to the user selection (CPU-specific blocks)
- Adoption of all the entries for SFCs or SFBs from the symbol table of the SINAUT TD7 library in all CPU modules involved if they do not already exist there.
- Synchronization of the time stamps of the compiled blocks with those of the library blocks to prevent time stamp conflicts in the TD7 section.

The compilation functions described create the corresponding ST7 blocks as a source file and store the source file in the *Sources* directory of the relevant CPU. The source file created in this way is then compiled and the blocks entered in the block directory.

Note

Only for installations with
- *SINAUT TD7 library* lower than version 2.0 and
- *SINAUT configuration software* lower than version 2.0:

If you want to change the STEP 7 user interface to English (in the SIMATIC Manager, *Options/Settings* menu), the *Mnemonic* option must not be changed to IEC! This must remain set to SIMATIC. Otherwise there will be compilation errors when generating the SINAUT program!

Dynamic data

All dynamic data required by the SINAUT TD7 software for the CPU is stored in the following data blocks:

- 1 DB BasicData
- n communication DBs
- m SMS data DBs

The quantity and numbers of the DBs are preset by the system. The numbers of the communication DBs as well as the quantity and numbers of the SMS data DBs can be modified in the *DB configuration* or *SMS configuration* tabs of the *Properties of subscriber* dialog that can be opened with the *Properties* context menu of the CPU modules in the subscriber list.

Basic blocks

When you compile for a CPU, a check is made to determine whether all the blocks are present that are required for the SINAUT ST7 software to run. The basic blocks are:

Table 1- 8 Basic blocks of TD7onCPU

Block name	Block type	Default block number
SubscriberObject	UDT	UDT127
ConnectionDescription	UDT	UDT126
Bcom	FB	FB127
Xcom	FB	FB126
Pcom	FB	FB125
SMS_Ctrl *	FB	FB124
BasicTask	FC	FC127
Search	FC	FC126
Diagnose	FC	FC125
Distribute	FC	FC124
Create	FC	FC123
Startup	FC	FC122

* Only if SMS messages were configured.

For each block there is a check to determine the following

- whether a symbol table entry exists for it and if it does
- whether the block exists in the user program

If there is no symbol table entry, a free block number for the relevant type is searched for, a suitable entry made in the symbol table and the block is entered in the source file for the current CPU.

If a symbol table entry already exists, but the corresponding block does not exist or only in a version lower than or equal to the library version, there is only an entry made in the source file.

In both cases, the compilation results in an executable TD7 program in the block directory. The selected sequence also ensures that any blocks that have been removed are always added again.

CPU-specific blocks

For all the blocks required dependent on the configuration, users can have them copied to the relevant CPU automatically from the SINAUT TD7 master source file.

This function is controlled by the symbol table in the relevant CPU. The user enters the required SINAUT TD7 blocks there. If, during compilation, it is determined that

- a symbol table entry exists for a single block,
- this block, however, does not exist in the user program of the CPU,

This block will be included in the generated source file.

The user can use the symbol table as a list of requirements. This makes it unnecessary to put together the CPU program by copying source files or blocks.

You will find an overview of the blocks generated in this way either in the SINAUT TD7 documentation, in the SINAUT TD7 master source file or in the symbol table of the SINAUT TD7 library.

Further user activities

The user now only needs to call FC BasicTask in OB1 and FC Startup in OB100 to activate the basic functionality of SINAUT TD7.

Note

For more detailed information on this and setting parameters for the data messages, refer to the description of the *TD7onCPU* software package.

With the generation/compilation functions, the user can also completely recompile the SINAUT program with the block numbers stored in the symbol table by deleting all SINAUT blocks in the user program and recompiling the source file.

1.11.4 Creating SINAUT subscriber numbers as comments

To make the subscriber numbers that are important for SINAUT communication visible in the SIMATIC Manager or in the hardware configuration, subscriber administration allows you to enter the subscriber number of SINAUT-networked components in the comment field of the properties.

The function is activated by selecting the option *Subscriber number as comment for stations, CPUs and TIMs* in the *Options* dialog when you call the *Save* function in the *Options* dialog.

With the following subscribers networked over SINAUT, the subscriber number is entered in the comment bar of the SIMATIC Manager:

- Stations:
- CPU modules

- TIM modules
- Third-party stations

The comments are visible in the SIMATIC Manager when you select the *View / Details* menu. The TIM modules are visible after expanding the tree structure and selecting a station.

Note

Creating the subscriber number when generating the system data overwrites comments previously entered in the network configuration without any possibility of restoring them.

1.11.5 Consistency check

The consistency check is always started automatically before you use the generate/compile functions in subscriber administration to prevent SDBs or DBs being created with inconsistent data.

The consistency check can also be started as a separate function using the *SINAUT / Check consistency...* menu in subscriber administration.

Errors detected during the consistency check are displayed to the user in an error list.

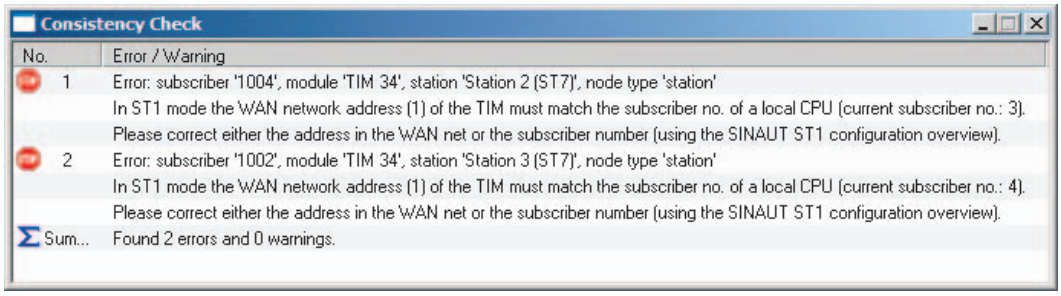

Consistency Check
No.
Error / Warning
1
Error: subscriber '1004', module 'TIM 34', station 'Station 2 (ST7)', node type 'station'
In ST1 mode the WAN network address (1) of the TIM must match the subscriber no. of a local CPU (current subscriber no.: 3).
Please correct either the address in the WAN net or the subscriber number (using the SINAUT ST1 configuration overview).
2
Error: subscriber '1002', module 'TIM 34', station 'Station 3 (ST7)', node type 'station'
In ST1 mode the WAN network address (1) of the TIM must match the subscriber no. of a local CPU (current subscriber no.: 4).
Please correct either the address in the WAN net or the subscriber number (using the SINAUT ST1 configuration overview).
Sum...
Found 2 errors and 0 warnings.

Figure 1-80 Example of an error list after running the consistency check

If inconsistent connections are found, the error list indicates that cause will be diagnosed in the connection configuration in the *Invalid connections* dialog.

After eliminating an error, you must save prior to the next consistency check otherwise be eliminated error will still be reported.

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

1.12 Change matrix

The change matrix describes the necessary follow-up activities of the user following typical actions in the SINAUT configuration tool.

Table 1- 9 Change matrix

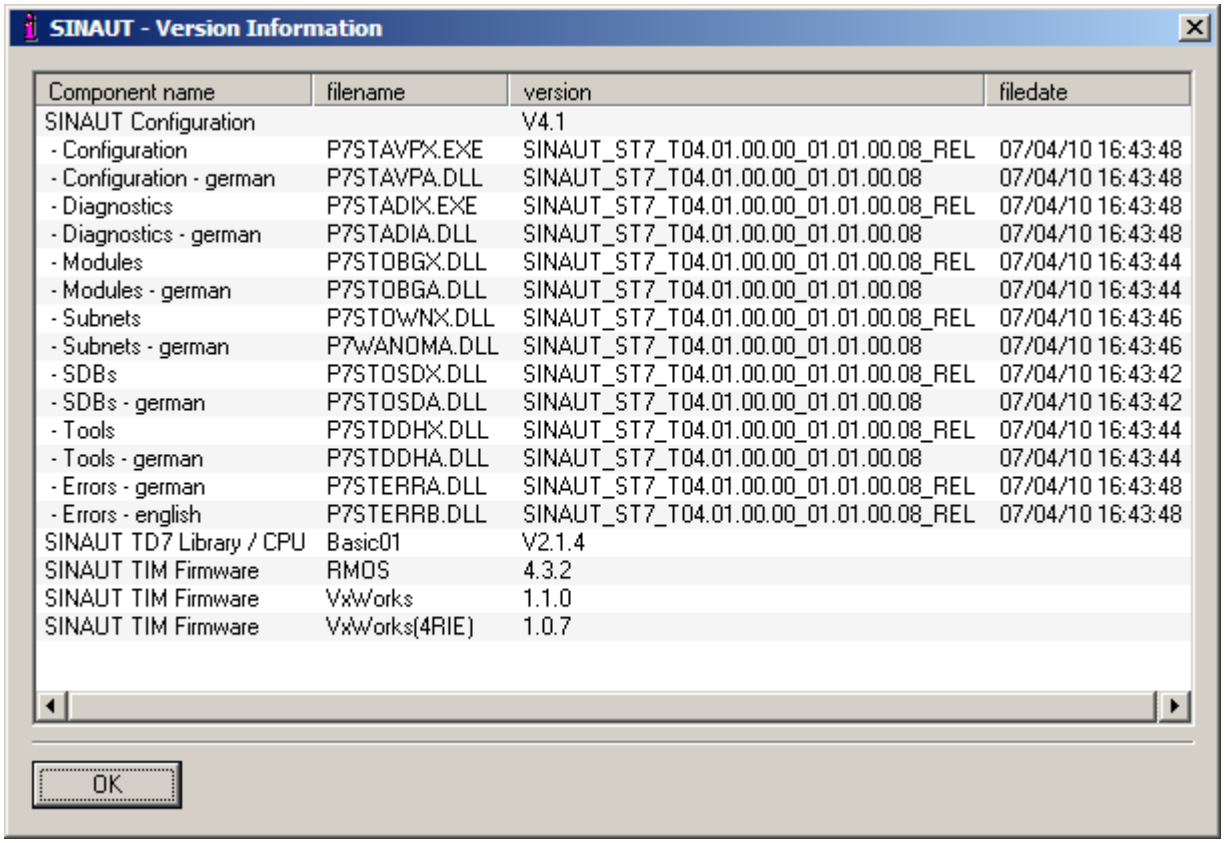
Object affected	Operator activity in the SINAUT configuration tool	Necessary follow-up action
Station	Adding a station	-
Station	Renaming a station	-
Station	Changing the parameter assignment of a station	-
Station	Deleting a station	All SINAUT connections running over a module in this station are then invalid and are removed the next time you open the SINAUT connection configuration. When necessary, these must be replaced by alternative connections. The SDBs or DBs of all modules that were involved in these deleted connections must be regenerated in the subscriber administration.
TIM module	Adding a TIM module	-
TIM module	Renaming a TIM module	-
TIM module	Changing the parameter assignment of a TIM module	If the parameters of a TIM module are changed, the SDBs must be regenerated in subscriber administration only for this TIM. Exception: If parameters are changed in the "WAN access" tab, this affects all SINAUT connections running over the modified WAN driver.
TIM module	Deleting a TIM module	All SINAUT connections running over this module are then invalid and are removed the next time you open the SINAUT connection configuration. When necessary, these must be replaced by alternative connections. The SDBs or DBs of all modules that were involved in these deleted connections must be regenerated in the subscriber administration.
Network	Adding a network	-
Network	Renaming a network	-
Network	Changing the parameter assignment of a network	The SDBs of all the modules connected to this network must be regenerated in the subscriber administration. In WAN networks, these are only TIM modules, in LANs all connected modules.
Network	Deleting a network	All SINAUT connections running over this network are then invalid and are removed the next time you open the SINAUT connection configuration. When necessary, these must be replaced by alternative connections. The SDBs or DBs of all modules that were involved in these deleted connections must be regenerated in the subscriber administration.
Network nodes	Adding a network node	-
Network nodes	Renaming a network node	-
Network nodes	Changing the parameter assignment of a network node	The SDBs of the module containing this network node must be regenerated. Exception: When changing the telephone number, the SDBs of all TIMs in the dial-up network must be regenerated.
Network nodes	Networking a network node	-

Object affected	Operator activity in the SINAUT configuration tool	Necessary follow-up action
Network nodes	Deleting a network node	All SINAUT connections running over this network node are then invalid and are removed the next time you open the SINAUT connection configuration. When necessary, these must be replaced by alternative connections. The SDBs or DBs of all modules that were involved in these deleted connections must be regenerated in the subscriber administration.
Network nodes	Canceling the networking of a network node	All SINAUT connections running over this network node are then invalid and are removed the next time you open the SINAUT connection configuration. When necessary, these must be replaced by alternative connections. The SDBs or DBs of all modules that were involved in these deleted connections must be regenerated in the subscriber administration.
SINAUT connection	Adding a SINAUT connection	The SDBs or DBs of all subscribers over which this connection runs must be regenerated in the subscriber administration.
SINAUT connection	Deleting a SINAUT connection	The SDBs or DBs of all subscribers over which the deleted connection runs must be regenerated in the subscriber administration.
SINAUT subscriber	Changing the parameter assignment of a SINAUT subscriber	The SDBs or DBs of all connections that run over this subscriber must be regenerated in the subscriber administration.
Destination subscriber properties of TD7onTIM	Changing the parameter assignment of a destination subscriber of TD7onTIM	The SDBs of all TIMs with TD7onTIM that communicate with this destination subscriber must be regenerated in subscriber administration.

1.13 Version information

The version information wizard displays the currency installed version and compilation time of the most important components of the SINAUT configuration software. The installed versions of the SINAUT TD7 library and the SINAUT TIM firmware are also shown.

This function is started from the Windows start menu *SIMATIC / SINAUT ST7 / Information*.



Component name	filename	version	filedate
SINAUT Configuration		V4.1	
- Configuration	P7STAVPX.EXE	SINAUT_ST7_T04.01.00.00_01.01.00.08_REL	07/04/10 16:43:48
- Configuration - german	P7STAVPA.DLL	SINAUT_ST7_T04.01.00.00_01.01.00.08	07/04/10 16:43:48
- Diagnostics	P7STADIX.EXE	SINAUT_ST7_T04.01.00.00_01.01.00.08_REL	07/04/10 16:43:48
- Diagnostics - german	P7STADIA.DLL	SINAUT_ST7_T04.01.00.00_01.01.00.08	07/04/10 16:43:48
- Modules	P7STOBGX.DLL	SINAUT_ST7_T04.01.00.00_01.01.00.08_REL	07/04/10 16:43:44
- Modules - german	P7STOBGA.DLL	SINAUT_ST7_T04.01.00.00_01.01.00.08	07/04/10 16:43:44
- Subnets	P7STOWNX.DLL	SINAUT_ST7_T04.01.00.00_01.01.00.08_REL	07/04/10 16:43:46
- Subnets - german	P7WANOMA.DLL	SINAUT_ST7_T04.01.00.00_01.01.00.08	07/04/10 16:43:46
- SDBs	P7STOSDX.DLL	SINAUT_ST7_T04.01.00.00_01.01.00.08_REL	07/04/10 16:43:42
- SDBs - german	P7STOSDA.DLL	SINAUT_ST7_T04.01.00.00_01.01.00.08	07/04/10 16:43:42
- Tools	P7STDDHX.DLL	SINAUT_ST7_T04.01.00.00_01.01.00.08_REL	07/04/10 16:43:44
- Tools - german	P7STDDHA.DLL	SINAUT_ST7_T04.01.00.00_01.01.00.08	07/04/10 16:43:44
- Errors - german	P7STERRA.DLL	SINAUT_ST7_T04.01.00.00_01.01.00.08_REL	07/04/10 16:43:48
- Errors - english	P7STERRB.DLL	SINAUT_ST7_T04.01.00.00_01.01.00.08_REL	07/04/10 16:43:48
SINAUT TD7 Library / CPU	Basic01	V2.1.4	
SINAUT TIM Firmware	RMOS	4.3.2	
SINAUT TIM Firmware	VxWorks	1.1.0	
SINAUT TIM Firmware	VxWorks(4RIE)	1.0.7	

Figure 1-81 SINAUT ST7 version information

1.14 Configuration practice

1.14.1 Downloading data blocks to the CPU

To download data blocks to the CPU module, you use the STEP 7 SIMATIC Manager standard tool. This allows you to copy blocks using drag-and-drop or a menu either in an online window (configured online access) or in the window with the accessible nodes (non-configured online access).

⚠ WARNING

When you download blocks to the automation system by dragging and dropping, you yourself are responsible for ensuring that the blocks are copied to the correct online object (in other words, the object with the correct MPI address). The STEP 7 tool does not check this.

For more detailed information on these activities, refer to the online help of the SIMATIC Manager.

1.14.2 Downloading system data blocks to the TIM

You should only download system data blocks (SDBs) to the TIM in the SIMATIC Manager or in the SINAUT Diagnostics and Service tool.

Note

When downloading system data blocks in *hardware configuration*, make sure that *no* connection SDBs (SDB7xx) are downloaded. If SDBs of this type need to be downloaded to the TIM module, you must use the SIMATIC Manager or the SINAUT Diagnostics and Service tool.

SDBs can also be downloaded in *network configuration*. Creating SDBs during network configuration is a different procedure from that in the SIMATIC Manager and hardware configuration and is not suitable for the TIM module. Copying SDBs to TIM modules should therefore only be done in the SIMATIC Manager or SINAUT Diagnostics and Service tool.

In the SIMATIC Manager, all SDBs of a module are indicated by a symbol with the name *System data*. This means that you can only ever manipulate all SDBs of a module as a single unit. Otherwise, the same applies as for data blocks.

In the hardware configuration, it is possible to download the SDBs of individual modules or entire stations.

In both cases, the function is followed by a dialog in which you are asked whether you want to restart the TIM module. This dialog must be exited with *Yes* to restart the TIM and activate the new SDBs.

1.14.3 Uploading stations with the Upload Station to PG function

The STEP 7 function *Upload PLC/Station to PG* allows the configuration of a connected station to be adopted. In conjunction with TIM modules, this function can only be used with certain restrictions.

- If the station to be uploaded is a TIM rack; in other words if the rack contains only stand-alone TIM modules, it is not possible to upload the station. The *Upload Station to PG* function can only be used in racks with CPU modules.
- If there is a 300 series CPU in the rack, the configuration can be uploaded, the TIM module represented in the rack is, however, not fully initialized and is not suitable for further configuration. This must be replaced by a new module from the hardware catalog.

⚠ WARNING

If the TIM modules uploaded in this way are further configured, problems can arise particularly in SINAUT communication and when handling the relevant module.

1.14.4 Changing the MPI address of the CPU

In hardware configuration, it is possible to change the MPI address of the CPU. If TIM modules are installed in the same rack, when downloading the SDBs ,**it is necessary** that the download is performed in two steps.

1. Download SDBs only to the CPU not to the TIM. Once the CPU module has received its new MPI address, the TIM modules go through a reset.
2. Download the SDBs to the TIM modules when they have completed the restart.

1.14.5 Copying projects in the SIMATIC Manager

In the SIMATIC Manager, you can, the entire projects by saving them under a different name (*File / Save As...* menu). After selecting the function, the *Save project as* dialog opens in which you enter the name and storage path.

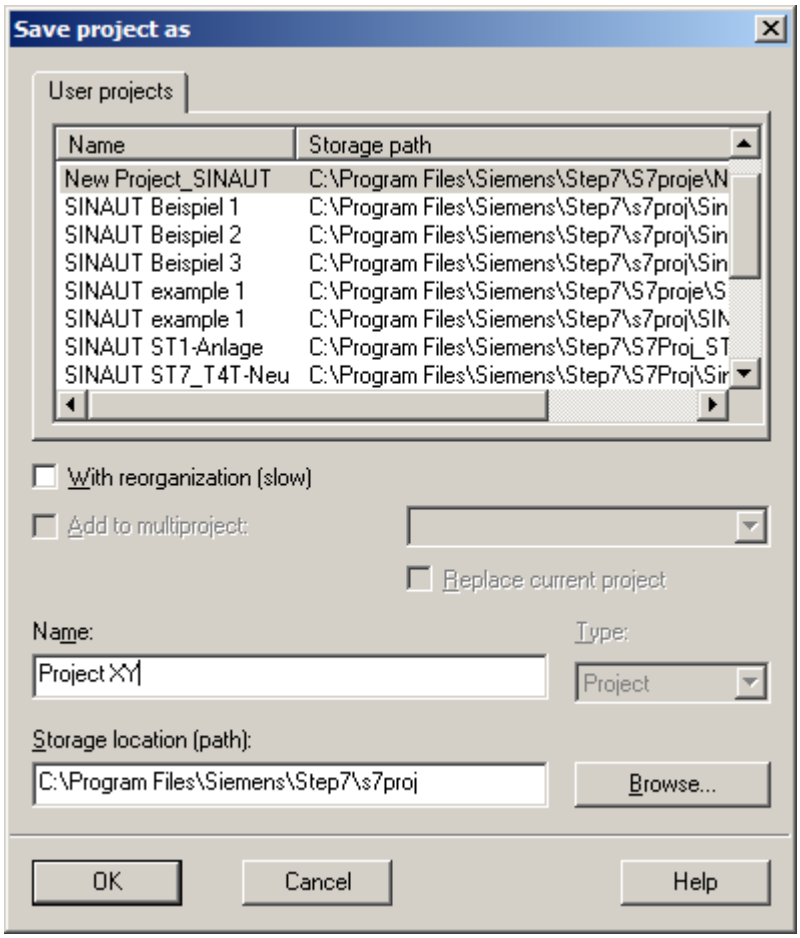


Figure 1-82 *Save project as* dialog for copying objects

When you save, you can also select the *With reorganization* copy option.

WARNING

When you copy projects with reorganization, the SINAUT subscribers and connection data are not copied. These must be reconfigured following the copy function.

If you copy projects without reorganization, the SINAUT data is also copied. There are no disadvantages of using this variant.
The *without reorganization* variant is preferable for SINAUT users.

1.14.6 Avoiding time stamp conflicts

Each block has an interface time stamp that provides information about when the interface visible to other blocks was last changed.

These time stamps are compared by the block editor when a block is opened. If inconsistencies are detected, the inconsistent calls are opened up; in other words, instead of a CALL command, the user finds the code generated by the system as a substitute for the CALL. This status is reported as a *time stamp conflict*. The substitute code must then be deleted by the user and the CALL parameters set again.

Inconsistencies of this type can occur between:

- OB and FBs / FCs
- FBs /FCs and FBs /FCs
- FBs and their instance DBs

Recommended procedure

- When using the configuration tool, this makes sure that the interface time stamp of all blocks regenerated by the tool (including instance data blocks) are synchronized with the interface time stamps of the blocks in the TD7 library.
- The time stamp of the block interfaces (supplied version / updates) are frozen until there is an actual change to an interface; in other words changes or corrections in the program code do not affect the interface time stamp.
- This procedure means that an update is possible at any time without causing a time stamp conflict.
- Directly inserting blocks by dragging them from the SINAUT library is permitted as long as the standard ST7 block number range (compare symbol table of the library) is not changed. This procedure is recommended

Not recommended procedure

- If the *TD7_UserSource* provided by the tool is compiled manually into a CPU block directory in which there are not yet any SINAUT TD7 blocks or in which some are missing, these blocks are given an interface time stamp that differs from that in the SINAUT ST7 library.
- Any SINAUT ST7 user programs already stored there then develop time stamp conflicts or they occur at the latest after the first update.
- Since SINAUT ST7 support is oriented mainly on the interface time stamp of the blocks due to version maintenance and management of upgrades, this procedure is generally not advisable.

Restrictions

The time stamps of the blocks that call SINAUT blocks are not synchronized. Here, time stamp conflicts can continue to occur.

The following sequence is therefore advisable for the user:

1. Before starting generation in the SINAUT ST7 configuration tool, a source file with all blocks that call SINAUT blocks is created.
2. The generation is run.
3. The previously generated source file is compiled.

SINAUT TD7 software package for the CPU 2

2.1 Overview

Introduction

For SINAUT ST7, there is the SINAUT TD7 software package for process data transmission between SINAUT subscribers over WAN, MPI bus and Ethernet. There are two variants of the software package:

- The *SINAUT TD7 for the CPU* software package, known as TD7onCPU, is a software package that has parameters assigned on the CPU and that runs on the CPU. It is used in all SINAUT stations in which TIM modules of the type TIM 3 or TIM 4 are configured.
- The *SINAUT TD7 for the TIM* software package, known as TD7onTIM, is a software package that is configured on Ethernet TIMs and runs on these TIMs. It can be used as an alternative to the TD7onCPU software package if an Ethernet TIM is configured in a SINAUT station.
  TD7onTIM is described in the section *Configuration software for SINAUT ST7*.

This chapter only describes the TD7onCPU software package.

The TD7 software package for the CPU contains blocks specifically for CPU modules. The package was designed so that it can run both on an S7-400 and on and S7-300 CPU. Exceptions to this will be pointed out explicitly.

With the aid of the SINAUT TD7 software, the user creates a program for the CPUs. This program allows change-driven transmission of process data between the individual CPUs and the control center, for example ST7cc. Failure of connections, CPUs, or the control center are displayed. Once a problem has been corrected or the CPUs/control center has started up, data is updated automatically.

Apart from process data transmission over WANs, the package is also suitable for local communication between CPUs if these are connected together over MPI. Even here, the local connections and CPUs are constantly monitored and data is updated automatically following startup or after a problem has been eliminated.

Note

Data communication from CPU to CPU over a WAN connection can only be implemented with the SINAUT TD7 software. This is not possible with the S7 communication SFBs/SFCs for configured and unconfigured connections. These are suitable only for local communication without a gateway.

The content of the SINAUT TD7 software package

The essential components are the package are as follows:

- Basic and auxiliary blocks
  Most of these blocks are always required on the CPU. A few are purely optional. The basic blocks handle central tasks such as startup, monitoring of connections and

availability of connection partners, general requests, time management, handling communication etc. The auxiliary blocks enter messages in the send buffer or fetch them from the receive buffer, handle send and receive jobs for specific connections, provide information as a result of searches, etc.

- Data point typicals
  These blocks are included in the CPU program depending on the type and amount of data to be transferred. They put messages together when data changes or when requested and output received process data.

Blocks required with the SINAUT ST7 configuration tool

To operate correctly, the TD7 package requires several data blocks per CPU. These DBs are generated automatically by the SINAUT ST7 configuration tool when the SINAUT connections are configured and they are stored on the relevant CPU. These are as follows:

- Central records DB
  This contains all the centrally required data including the records of all communication partners and the connections to be managed.
- Communication DBs
  A separate communication DB is created for each connection. This DB contains a send and receive buffer and all the data required for controlling and monitoring the connection.

Along with the data blocks mentioned above, the SINAUT ST7 configuration tool also stores the basic and auxiliary blocks necessary for a functioning program in the CPU program directory. Users only need to copy the required data point typicals from the SINAUT TD7 library to the CPU program directory and they can then create the SINAUT user program.

Note

The blocks stored in the CPU program director by the SINAUT ST7 configuration tool exist once as individual blocks in the 'Blocks' directory and also in the 'Sources' directory as an STL source file with the name 'TD7_UserSource'.

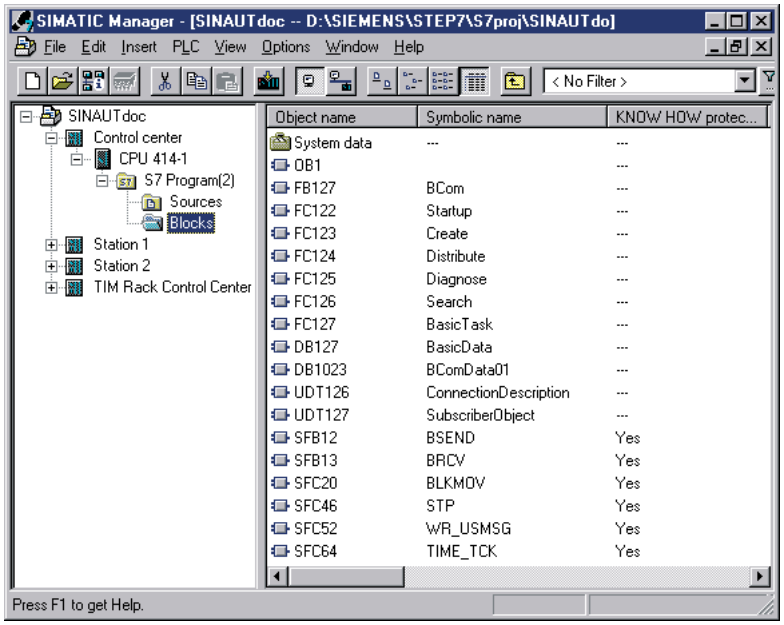


Figure 2-1 Basic and auxiliary blocks in the blocks program directory

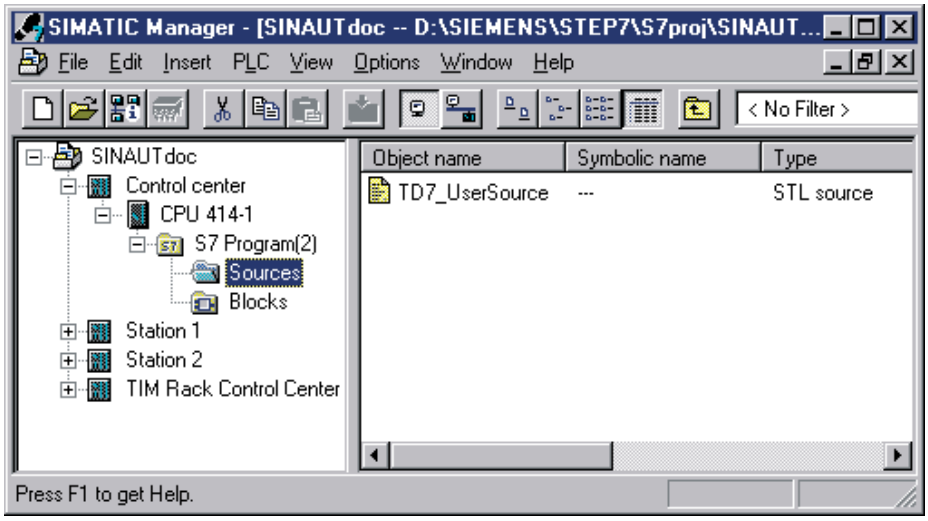


Figure 2-2 TD7_UserSource im Programmverzeichnis Quellen

Note

If you want to change the STEP 7 user interface to English (in the SIMATIC Manager, Options/Settings menu), the following applies depending on the version of the SINAUT configuration tool:

- SINAUT configuration tool version < 2.0
  The *Mnemonic* option must not be changed to IEC or English! This must remain set to SIMATIC. Otherwise there will be compilation errors when generating the SINAUT program!
- SINAUT configuration tool as of version 2.0
  The *Mnemonic* option can be changed to IEC or English! The configuration tool checks the setting for Mnemonic and compiles the SINAUT program correctly.

Basic structure of the user program

The SINAUT user program consists of the following:

- Startup program OB100
  Here, only the startup block for ST7 "FC Startup" needs to be called. It does not require any parameters.
- Cyclic program OB1
  In the simplest case, this program consists of the basic block "FC BasicTask". Following this, the user calls all the data point typicals required for the CPU and sets the parameters to suit the particular application.
- Time-driven program OB35 (or another cyclic interrupt OB)
  This program is required only when counted pulses need to be acquired on the CPU. FC PulseCounter is called once or more than once in the cyclic interrupt OB.

2.1.1 SINAUT TD7 Library

Introduction

When you install the SINAUT software, a library is also created with the SINAUT TD7 software package. This is access in the same way as all other libraries in STEP 7; in other words, from the SIMATIC Manager, you can display a list of available libraries with the *Open* menu in the *Libraries* tab. The SINAUT library is located there under the name *SINAUT TD7 Library*.

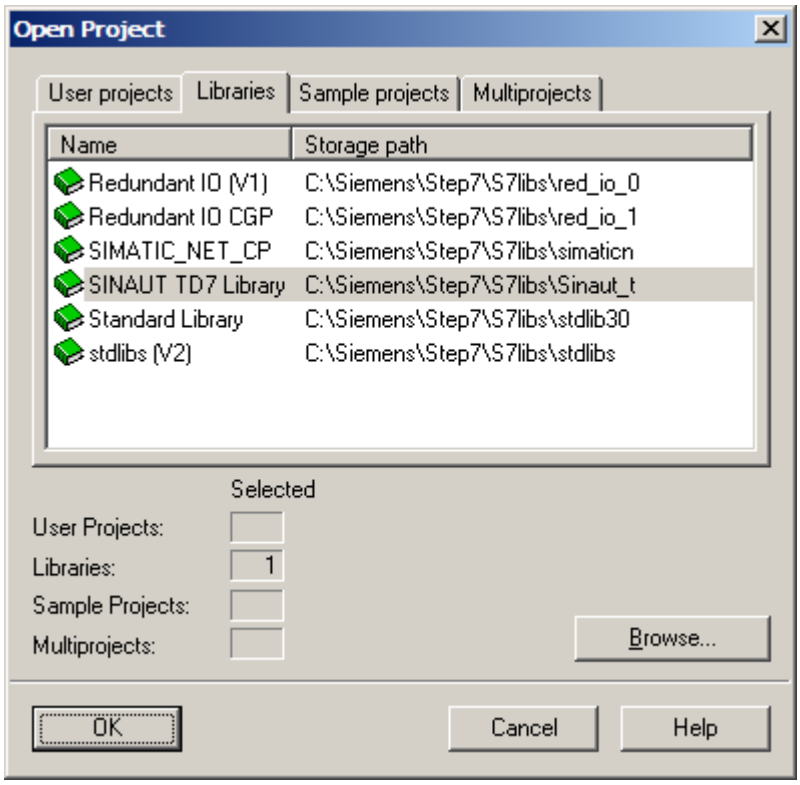


Figure 2-3 Opening the SINAUT library *SINAUT TD7 Library*

Structure of the library

SINAUT TD7 Library contains the directory *Basic01* that is made up of the following sections:

- Source
  Contains the two STL source files Basic01_Source_de (blocks with German mnemonics) and Basic01_Source_en (blocks with English mnemonics). This contains all the blocks of the TD7 library in STL source format.
- Blocks
  contains all the blocks of the TD7 library in block format. This contains all the SINAUT blocks of the type FB, FC, DB, UDT and VAT and the SIMATIC system blocks SFC and SFB that are used by the SINAUT software.

- Symbols
  Contains the symbol table of the SINAUT TD7 library.

Note

Never modify the content of *Source*, *Blocks* or *Symbols* in the SINAUT TD7 library! Event renaming or moving the library to a different directory is prohibited!

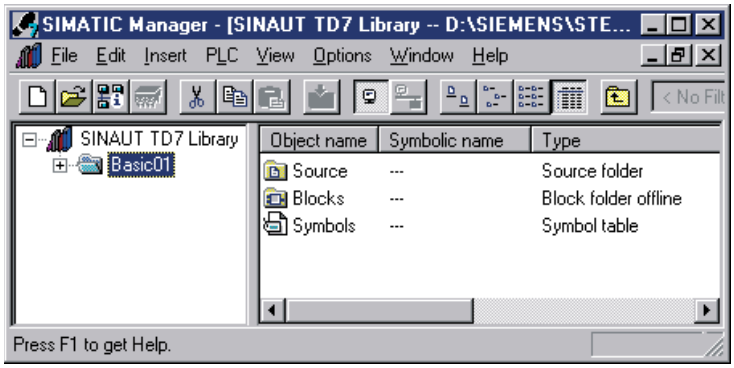


Figure 2-4 Components of the SINAUT TD7 library

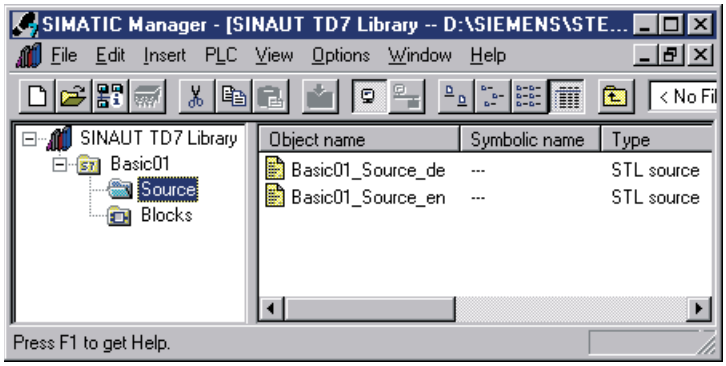


Figure 2-5 Source STL files *Basic01_Source_de/_en* in the SINAUT TD7 library

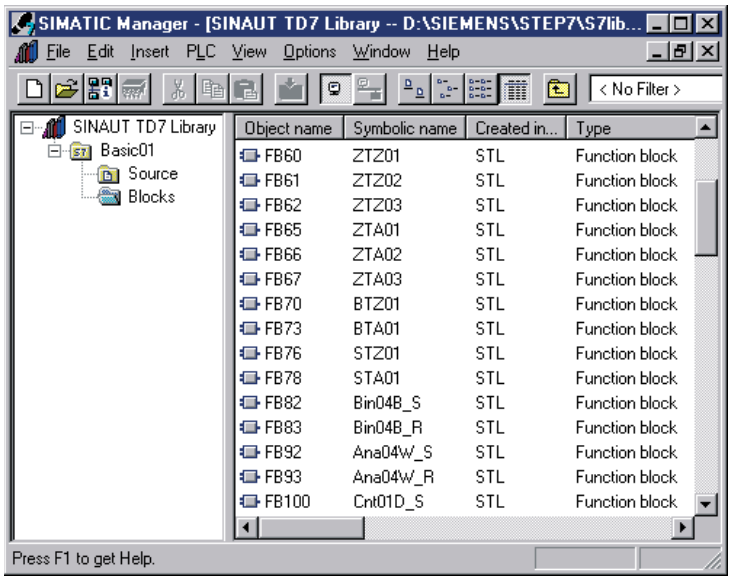

SIMATIC Manager - [SINAUT TD7 Library -- D:\SIEMENS\STEP7\S7lib...
File Edit Insert PLC View Options Window Help
< No Filter >
SINAUT TD7 Library
Basic01
Source
Blocks
Object name | Symbolic name | Created in... | Type
FB60 ZTZ01 STL Function block
FB61 ZTZ02 STL Function block
FB62 ZTZ03 STL Function block
FB65 ZTA01 STL Function block
FB66 ZTA02 STL Function block
FB67 ZTA03 STL Function block
FB70 BTZ01 STL Function block
FB73 BTA01 STL Function block
FB76 STZ01 STL Function block
FB78 STA01 STL Function block
FB82 Bin04B_S STL Function block
FB83 Bin04B_R STL Function block
FB92 Ana04W_S STL Function block
FB93 Ana04W_R STL Function block
FB100 Cnt01D_S STL Function block
Press F1 to get Help.

Figure 2-6 Blocks of the SINAUT TD7 library in block format

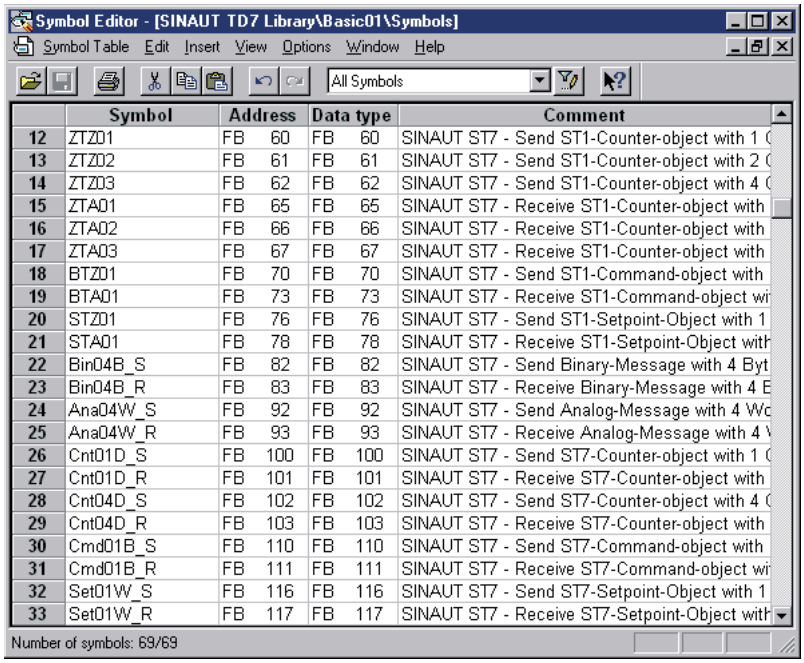

Symbol Editor - [SINAUT TD7 Library\Basic01\Symbols]
Symbol Table Edit Insert View Options Window Help
All Symbols
Symbol | Address | Data type | Comment
12 ZTZ01 FB 60 FB 60 SINAUT ST7 - Send ST1-Counter-object with 1
13 ZTZ02 FB 61 FB 61 SINAUT ST7 - Send ST1-Counter-object with 2
14 ZTZ03 FB 62 FB 62 SINAUT ST7 - Send ST1-Counter-object with 4
15 ZTA01 FB 65 FB 65 SINAUT ST7 - Receive ST1-Counter-object with
16 ZTA02 FB 66 FB 66 SINAUT ST7 - Receive ST1-Counter-object with
17 ZTA03 FB 67 FB 67 SINAUT ST7 - Receive ST1-Counter-object with
18 BTZ01 FB 70 FB 70 SINAUT ST7 - Send ST1-Command-object with
19 BTA01 FB 73 FB 73 SINAUT ST7 - Receive ST1-Command-object wi
20 STZ01 FB 76 FB 76 SINAUT ST7 - Send ST1-Setpoint-Object with 1
21 STA01 FB 78 FB 78 SINAUT ST7 - Receive ST1-Setpoint-Object with
22 Bin04B_S FB 82 FB 82 SINAUT ST7 - Send Binary-Message with 4 Byt
23 Bin04B_R FB 83 FB 83 SINAUT ST7 - Receive Binary-Message with 4 B
24 Ana04W_S FB 92 FB 92 SINAUT ST7 - Send Analog-Message with 4 Wo
25 Ana04W_R FB 93 FB 93 SINAUT ST7 - Receive Analog-Message with 4
26 Cnt01D_S FB 100 FB 100 SINAUT ST7 - Send ST7-Counter-object with 1
27 Cnt01D_R FB 101 FB 101 SINAUT ST7 - Receive ST7-Counter-object with
28 Cnt04D_S FB 102 FB 102 SINAUT ST7 - Send ST7-Counter-object with 4
29 Cnt04D_R FB 103 FB 103 SINAUT ST7 - Receive ST7-Counter-object with
30 Cmd01B_S FB 110 FB 110 SINAUT ST7 - Send ST7-Command-object with
31 Cmd01B_R FB 111 FB 111 SINAUT ST7 - Receive ST7-Command-object wi
32 Set01W_S FB 116 FB 116 SINAUT ST7 - Send ST7-Setpoint-Object with 1
33 Set01W_R FB 117 FB 117 SINAUT ST7 - Receive ST7-Setpoint-Object with
Number of symbols: 69/69

Figure 2-7 The symbol table *Symbols* in the SINAUT TD7 library

Note

If you want to change the numbers of individual SINAUT blocks to avoid conflicts with blocks already used in the user program, you must create a simple table for the relevant CPU block(s) in which all the required SINAUT blocks per CPU are listed. This involve is not only the blocks whose number has been changed in the simple table but also those whose number remains unchanged. For more detailed information, refer to the section *Changing SINAUT block numbers*.

2.1.2 Block overview

Introduction

The following table lists all the blocks contained in the TD7 library. For each block, the table provides:

- The standard block number under which the lock is available in the TD7 library. This number can be changed when necessary.
- The symbolic name of the block. You can call the relevant block in the user program using this name.
- A note indicating whether this block is an auxiliary block that is called indirectly. This is important in case the standard number of the auxiliary block needs to be changed.
- A list of the auxiliary blocks required by the relevant block. Only the SINAUT auxiliary blocks are specified since only these blocks need to be transferred from the TD7 library into the user program directory (normally performed automatically by the ST7 configuration tool). The information on the auxiliary blocks is also important if one or more of the listed SINAUT auxiliary blocks are given a different block number. The block must then be recompiled.
- A brief explanation of the function of the block.

You will find the blocks for SINAUT ST1 in release 05/2007 of the manual.

Table 2- 1 SINAUT TD7 Library: Block overview

Block no.	Symbolic name	Aux. block	Required (SINAUT) auxiliary blocks	Explanation
Function blocks FB				
FB82	Bin04B_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 status message object, send 4 bytes of status/binary information.
FB83	Bin04B_R	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 status message object, receive 4 bytes of status/binary information.
FB84	Dat12D_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 data object, send max. 12 double words with any information.
FB85	Dat12D_R	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 data object, receive max. 12 double words with any information.
FB92	Ana04W_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 analog value object, send 4 analog values (16-bit value in the INT format).
FB93	Ana04W_R	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 analog value object, receive 4 analog values (16-bit value in the INT format).
FB100	Cnt01D_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 counted value object, send 1 counted value (32-bit ST1 format).
FB101	Cnt01D_R	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 counted value object, receive 1 counted value (32-bit ST1 format).
FB102	Cnt04D_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 counted value object, send 4 counted values (32-bit ST1 format).
FB103	Cnt04D_R	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 counted value object, receive 4 counted values (32-bit ST1 format).

Block no.	Symbolic name	Aux. block	Required (SINAUT) auxiliary blocks	Explanation
FB110	Cmd01B_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 command object, send 1 byte commands (1-out-of-8 ST1 format).
FB111	Cmd01B_R	-	DB127 BasicData, FC126 Search	ST7 command object, receive 1 byte commands (1-out-of-8 ST1 format).
FB116	Set01W_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 setpoint object, send 1 setpoint (16 bits) and receive current local setpoint.
FB118	Par12D_S	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 parameter object, send max. 12 double words with parameters and receive current local parameters.
FB119	Par12D_R	-	DB127 BasicData, FC123 Create, FC126 Search	ST7 parameter object, receive max. 12 double words with parameters and send current local parameters.
FB124	SMS_Control	-	DB127 BasicData, FC123 Create, FC125 Diagnose, FC126 Search	Block for sending SMS messages.
FB125	PCom	Yes	DB127 BasicData, FC124 Distribute, FC125 Diagnose, FC126 Search	Block for communication over a peripheral bus connection (SFCs WR_REC and RD_REC are used).
FB126	XCom	Yes	DB127 BasicData, FC124 Distribute, FC125 Diagnose, FC126 Search	Communication block for an unconfigured X connection (SFCs X_SEND and X_RCV are used).
FB127	BCom	Yes	DB127 BasicData, FC124 Distribute, FC125 Diagnose, FC126 Search	Communication block for a configured communication block connection (SFBs BSEND and BRCV are used).
Functions FC				
FC113	PartnerMonitor	-	DB127 BasicData, FC125 Diagnose, FC126 Search	Allows the display of important status information and control for a SINAUT subscriber.
FC114	Trigger	-	DB127 BasicData, FC125 Diagnose	Sets an output at a defined time or at a defined interval.
FC115	PartnerStatus	-	DB127 BasicData, FC125 Diagnose, FC126 Search	Displays the connection status for up to 8 SINAUT subscribers.
FC116	Safe	-	DB127 BasicData	Block for saving command and setpoint input.
FC117	PulseCounter	-	DB127 BasicData	Block for acquiring up to 8 counter inputs
FC118	TestCopy	-	DB127 BasicData	Test block for logging received and/or sent messages.
FC119	ListGenerator300	-	DB127 BasicData	Block for generating an object list. Version for S7-300.
FC120	ListGenerator400	-	DB127 BasicData	Block for generating an object list. Version for S7-400.
FC121	TimeTask	-	DB127 BasicData, FC125 Diagnose	Block for keeping the date and time on a CPU
FC122	Startup	-	DB127 BasicData	Startup block.
FC123	Create	Yes	DB127 BasicData, FC125 Diagnose, FC126 Search	Block for creating messages and entering them in the send buffer.

Block no.	Symbolic name	Aux. block	Required (SINAUT) auxiliary blocks	Explanation
FC124	Distribute	Yes	DB127 BasicData, FC125 Diagnose, FC126 Search	Block for distributing received messages to the local destination objects.
FC125	Diagnose	Yes	DB127 BasicData	Block for entering system messages in the diagnostic buffer of the CPU.
FC126	Search	Yes	-	Block for handling searches.
FC127	BasicTask	-	DB127 BasicData, FC123 Create, FC125 Diagnose, FB125 PCom, FB126 XCom, FB127 BCom	Block for handling all SINAUT basic tasks on the CPU.
Data blocks DB				
DB99	TestCopyData	-	-	Data block for the test block FC TestCopy.
DB125	SMS_Data	-	UDT125 ShortMessageObject	Data block for FB SMS_Control for entry, for example of SMS texts.
DB127	BasicData	Yes	UDT126 ConnectionDescription, UDT127 SubscriberObject	Data block for entry of SINAUT basic information.
User-defined data types UDT				
UDT125	ShortMessage object	Yes	-	SMS object (for DB SMS_Data).
UDT126	Connection description	Yes	-	Connection description (for DB BasicData).
UDT127	SubscriberObject	Yes	-	Subscriber object (for DB BasicData).
Variable table VAT				
VAT99	VAT_TestCopy	-	-	Variable table for the test block FC TestCopy.
System function blocks SFB				
SFB12	BSEND	Yes	-	Block-oriented sending of data over a configured connection.
SFB13	BRCV	Yes	-	Block-oriented reception of data over a configured connection.
System functions SFC				
SFC0	SET_CLK	Yes	-	Set CPU clock.
SFC1	READ_CLK	Yes	-	Read CPU clock.
SFC20	BLKMOV	Yes	-	Copy variables.
SFC22	CREAT_DB	Yes	-	Create data block.
SFC23	DEL_DB	Yes	-	Delete data block.
SFC24	TEST_DB	Yes	-	Test data block.
SFC25	COMPRESS	Yes	-	Compress user memory.
SFC46	STP	Yes	-	Set CPU to STOP.
SFC52	WR_USMSG	Yes	-	Write user diagnostic message to the diagnostic buffer.
SFC64	TIME_TCK	Yes	-	Read system time.

Block no.	Symbolic name	Aux. block	Required (SINAUT) auxiliary blocks	Explanation
SFC65	X_SEND	Yes	-	Send data over an unconfigured connection.
SFC66	X_RCV	Yes	-	Receive data over an unconfigured connection.

2.1.3 Changing SINAUT block numbers

Introduction

The blocks in the SINAUT library have fixed block numbers that you should, whenever possible, use with these numbers in your user program. This applies to all SFB an SFC system blocks whose numbers cannot be modified. The numbers of SINAUT FBs, FCs, DBs and UDTs can, however, be adapted when necessary. The effort required for the changes varies from case to case. The three following modification stages can be distinguished:

1. Only numbers of SINAUT blocks are changed that do not belong to the category of auxiliary blocks (there is a dash in the 'Aux. block' column in the table),
   and
   The standard numbers of the SINAUT auxiliary blocks were not changed in the relevant project.
2. Only the numbers of the SINAUT UDTs are changed.
3. One or more SINAUT auxiliary blocks will be given a different number (in the table, there is a 'yes' in the 'Aux. block' column),
   or
   you want to copy for the SINAUT blocks into a project in which numbers of SINAUT auxiliary blocks have already been changed (it does not matter whether these are auxiliary blocks or not).

Depending on the change level, you should follow the steps outlined below:

Note

With each change to a block number made by the user, make sure that the number change is also entered in the symbol table of the CPU. Otherwise, errors will occur when you save in the 'Subscriber Administration' SINAUT tool.

Changing the numbers of blocks other than auxiliary blocks

The prerequisite for using the instructions here is as follows:

- You only want to change numbers of SINAUT blocks that do not belong to the auxiliary block category,
  and
- The standard numbers of the SINAUT auxiliary blocks have not been changed in the relevant project.

Initial situation

You have configured your SINAUT installation as already described in the section 'Configuration software for SINAUT ST7'; in other words, you have completed the connection configuration with the appropriate SINAUT configuration tool and have started to save in the 'Subscriber Administration' SINAUT tool. The result is that all the necessary SINAUT blocks already exist in all the program directories of the CPU in addition to the SDB directory and OB1 as shown in the following figure.

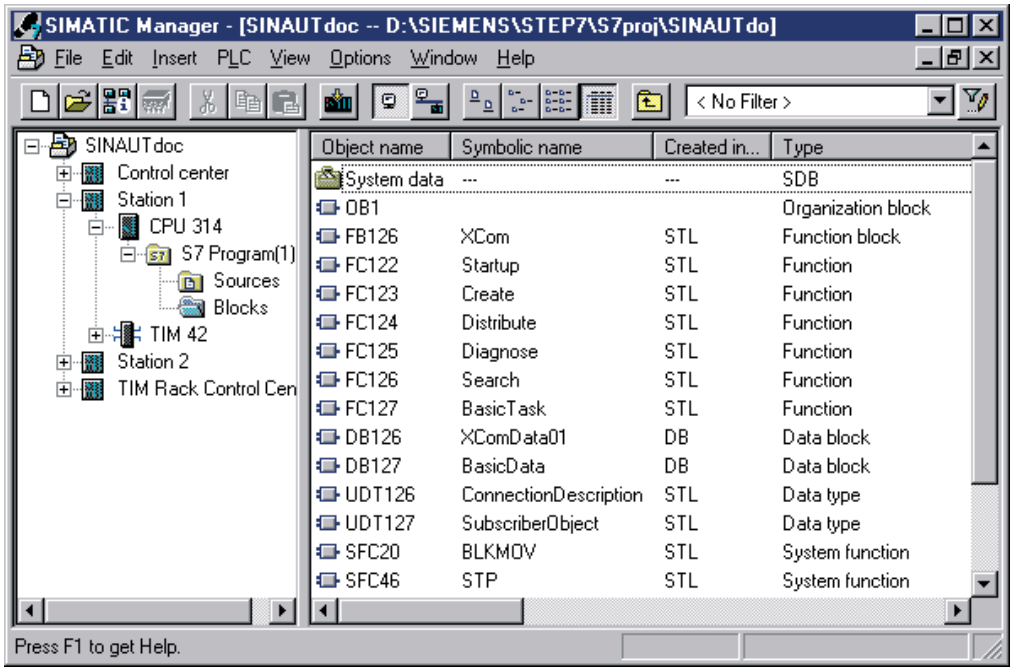

SIMATIC Manager - [SINAUTdoc -- D:\SIEMENS\STEP7\S7proj\SINAUTdo]
File Edit Insert PLC View Options Window Help
< No Filter >
SINAUTdoc
Control center
Station 1
CPU 314
S7 Program(1)
Sources
Blocks
TIM 42
Station 2
TIM Rack Control Cen
Object name | Symbolic name | Created in... | Type
System data --- --- SDB
OB1 Organization block
FB126 XCom STL Function block
FC122 Startup STL Function
FC123 Create STL Function
FC124 Distribute STL Function
FC125 Diagnose STL Function
FC126 Search STL Function
FC127 BasicTask STL Function
DB126 XComData01 DB Data block
DB127 BasicData DB Data block
UDT126 ConnectionDescription STL Data type
UDT127 SubscriberObject STL Data type
SFC20 BLKMOV STL System function
SFC46 STP STL System function
Press F1 to get Help.

Figure 2-8 Example of a project (SINAUT basic blocks already present)

Making the change

Copy the additional blocks that you require directly from the TD7 library to the program directory of the relevant CPU. If the number of one of the copied blocks is identical to an existing block, a dialog is displayed automatically indicating the number conflict. You will be asked whether or not you want to remain the block. After clicking on *Rename...*, a further dialog appears in which you can enter the new number, for example FB82 is renamed to FB8 as shown in the following figure.

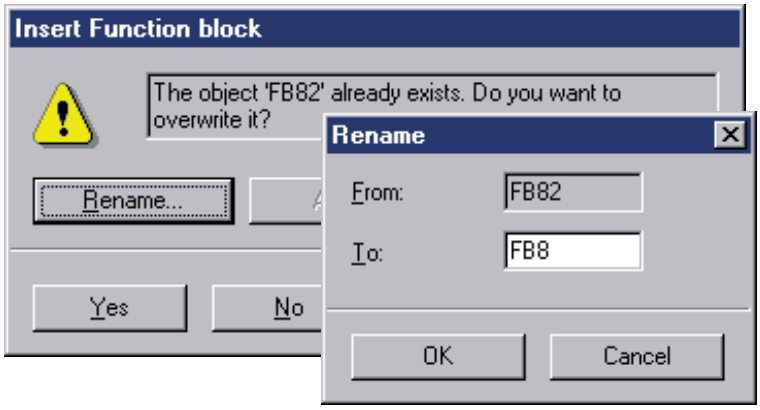


Figure 2-9 Renaming blocks in the dialog

If the block number is changed using this dialog, the number change is automatically included in the symbol table. This can be recognized because the renamed block in the block directory is displayed immediately with the corresponding name from the TD7 library.

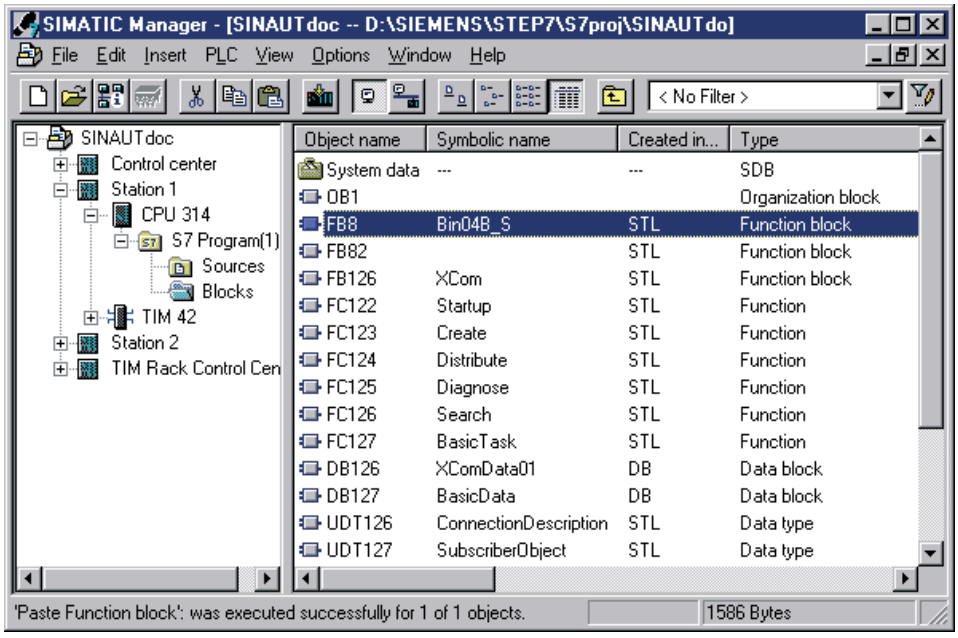

SIMATIC Manager - [SINAUTdoc -- D:\SIEMENS\STEP7\S7proj\SINAUTdo]
File Edit Insert PLC View Options Window Help
< No Filter >
SINAUTdoc
Control center
Station 1
CPU 314
S7 Program(1)
Sources
Blocks
TIM 42
Station 2
TIM Rack Control Cen
Object name | Symbolic name | Created in... | Type
System data --- --- SDB
OB1 Organization block
FB8 Bin04B_S STL Function block
FB82 STL Function block
FB126 XCom STL Function block
FC122 Startup STL Function
FC123 Create STL Function
FC124 Distribute STL Function
FC125 Diagnose STL Function
FC126 Search STL Function
FC127 BasicTask STL Function
DB126 XComData01 DB Data block
DB127 BasicData DB Data block
UDT126 ConnectionDescription STL Data type
UDT127 SubscriberObject STL Data type
'Paste Function block': was executed successfully for 1 of 1 objects.
1586 Bytes

Figure 2-10 Renaming with automatic symbol assignment

If you copy several blocks at one time, the dialog is displayed for each block whose number is already being used. You can therefore copy all the required blocks per CPU at one time and then adapt the block numbers and entries in the symbol table as necessary.

If the dialog does not appear when you copy blocks, there is no conflict with existing blocks. If you nevertheless want to give blocks a different number, you must change these numbers directly in the block directory and also adapt the relevant entries in the symbol table (although the symbolic name of a SINAUT block was entered automatically in the symbol table of the destination CPU when the blocks were copied from the TD7 library, the new numbers are not taken into account since they were not changed initially during copying).

If you want to change one of the blocks already stored by the SINAUT tool (here, this can only involve the blocks FC122 Startup and FC127 BasicTask because all other blocks

belong to the auxiliary block category), the numbers of these blocks can be changed directly in the block directory. Do not forget, however, to make the same number change in the symbol table of the CPU.

Note

Never use this method to change the numbers of the SINAUT data blocks. These are the data blocks with the symbolic names *BasicData*, *XComData01*, *XComData02* etc., *PComData01*, *PComData02* etc. and *BComData01*, *BComData02* etc.

If you want to make the same changes for more than one CPU, the most efficient method is to make the changes first in the block directory and the symbol table of one CPU. You can then copy the modified blocks from the block directory of the finished CPU to the other CPUs and then do the same for the symbol tables.

Changing the numbers of UDTs

The prerequisite for using the instructions here is as follows:

- You only want to change the numbers of SINAUT UDTs.

Initial situation

You have completed the configuration of your SINAUT installation including the connection configuration with the appropriate SINAUT configuration tool. You have not yet saved in the 'Subscriber Administration' SINAUT tool. The result is that there are not yet any SINAUT blocks in the program directories of the CPUs. The block directory CPU contents, for example, only the system data and OB1 as shown in the following figure.

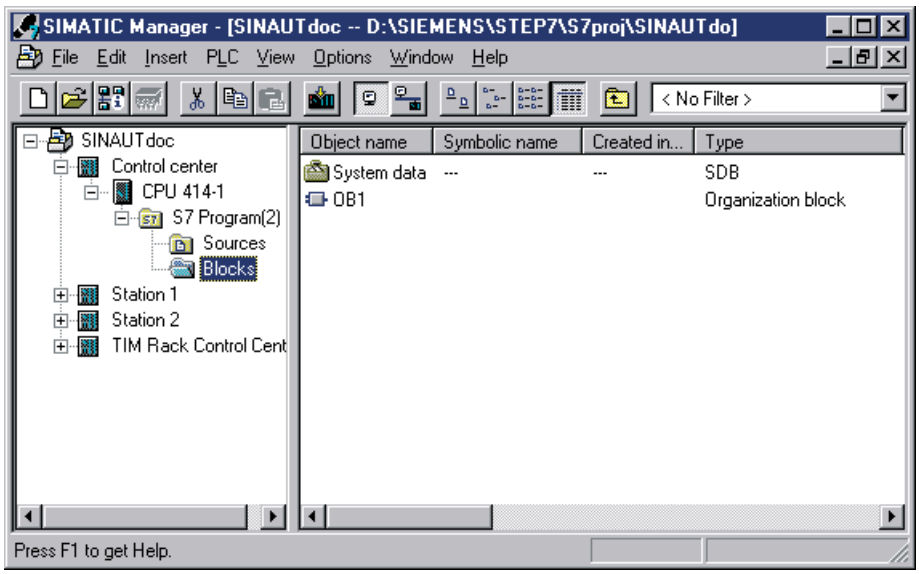


Figure 2-11 Example of a project (still without user or SINAUT program)

Making the change

The change to the UDT numbers is made indirectly using the symbol table of the CPU with the aid of the 'Subscriber Administration' SINAUT Configuration Tool.

Step 1

- Copy the UDTs that you want to change in your program from the symbol table of the TD7 library to the symbol table of the first CPU.
- Change the numbers of the SINAUT UDTs to the numbers you require in the symbol table.
- Save the symbol table.
- Then copy the rows with the changed UDT numbers to the symbol tables of all other CPUs of your project. Do not forget to save all the modified symbol tables.

Step 2

- Once all the symbol tables have the required values and have been saved, change to the 'Subscriber Administration' SINAUT ST7 Configuration Tool.
- Click the 'Save' button. An extra dialog 'Properties' opens (see following figure).
- Make sure that a check mark is entered in front of the following options in this dialog:
  - 'System data blocks for TIMS and CPUs'
    and
  - 'SINAUT TD7 blocks for CPUs'
  - and a dot in front of 'for all CPUs'. Then click on 'OK'.

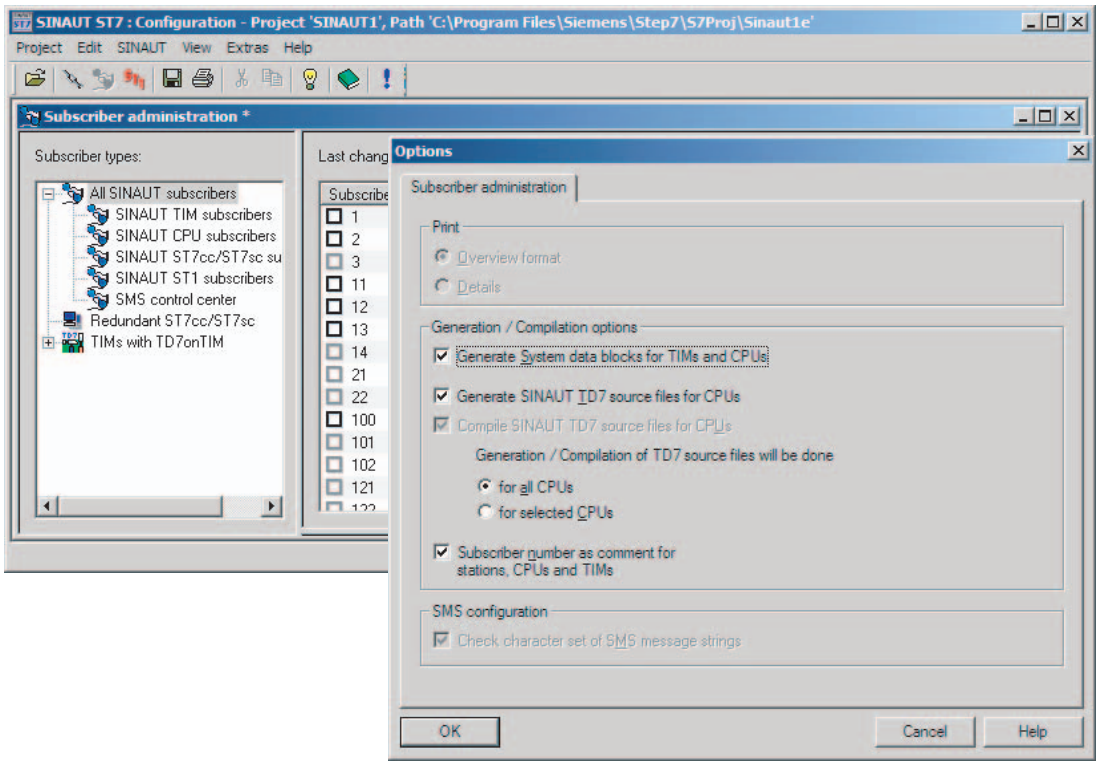


Figure 2-12 Dialog for triggering compilation of the SINAUT TD7 blocks

After you have saved, all the SINAUT blocks basically required are entered in the block directories of all CPUs, and possibly also the UDTs with the new numbers you have

selected. The SINAUT UDTs are required to generate the BasicData data block. This central administrative block also exists in the block directory and is generated taking into account the new UDT numbers.

Changing the numbers of auxiliary blocks

The prerequisite for using the instructions here is as follows:

- You want to change the numbers of SINAUT auxiliary blocks in a new project (see below, initial situation 1),
  or
- You want to copy further SINAUT blocks into a project in which numbers of SINAUT auxiliary blocks have already been changed; it does not matter whether these are auxiliary blocks or not (see below, initial situation 2).

Initial situation 1

You configure your SINAUT installation and as described in the section 'Configuration software for SINAUT ST7' and have configure the connections with the appropriate SINAUT configuration tool. You have not yet started the 'Subscriber Administration' SINAUT tool.

Or the S7 CPUs and TIM modules exist in your project. The block directory of the CPU contains only the system data and OB1.

Making the change

The change to the numbers is made indirectly using the symbol table of the CPU with the aid of the 'Subscriber Administration' SINAUT Configuration Tool.

Step 1

Fill the block directories of all S7 CPUs with the user blocks (FBs, FCs, DBs and UDTs) whose numbers you want to retain. If you have assigned symbolic names to your blocks, enter these in the symbol tables of the CPUs.

Step 2

Then check which SINAUT blocks you require for your program. The following table shows which blocks are always required by the SINAUT program (depending on the CPU type, there are slight variations). The SMSxxx blocks are required only when the SMS function is configured.

Table 2- 2 SINAUT blocks that are always required

S7-300	S7-400
FB126 XCom	FB127 BCom
FB125 PCom	
FB124 SMS_Control	
FC122 Startup	
FC123 Create	
FC124 Distribute	
FC125 Diagnose	

S7-300	S7-400
FC126 Search	
FC127 BasicTask	
DBxxx SMS_Data	
DB127 BasicData	
UDT126 ConnectionDescription	
UDT127 SubscribeObject	

Based on the table, you can see which other blocks may be required. At least one or more blocks for data acquisition and output are required. These are the blocks in the range from FB40 ... FB117. Further optional basic functions may also be required that are available in the library under FC114 ...FC121 (you will find information on these optional basic functions below in the section 'SINAUT startup program in OB100').

When putting together the required SINAUT blocks, does not matter whether or not these blocks have numbers that need to be changed. You should identify all the SINAUT blocks required per CPU.

Copy the rows from the symbol table of the SINAUT TD7 library with the required SINAUT blocks to the symbol tables of the individual CPUs. In the symbol tables of the CPUs, change the SINAUT block numbers to the required numbers.

Note

If a change is the same for several CPUs, the most efficient method is to adapt the symbol table first for one of the CPUs and then to copy these entries to the symbol tables of all other CPUs.

Step 3

Once all the symbol tables have the required values, change to the 'Subscriber Administration' SINAUT ST7 Configuration Tool. Click the 'Save' button. An additional dialog "Properties" opens. In this dialog, make sure that there is a check mark in front of 'System data blocks for TIMs and CPUs' and in front of 'SINAUT TD7 block for CPUS' and a dot in front of 'for all CPUs'. Then click on 'OK'.

After the save is completed, the SINAUT blocks listed in the symbol table have been added to the content of the block directory and one or more communication DBs have been added (recognizable by their symbolic names, for example XComData01 or BComData01). The SINAUT blocks now have their new numbers. If numbers of SINAUT auxiliary blocks have been changed, all SINAUT blocks that call these auxiliary blocks have been recompiled; in other words, they now call these auxiliary blocks internally with the new numbers.

Initial situation 2

You want to copy further SINAUT blocks to a CPU on which the numbers of SINAUT auxiliary blocks have already been changed. It does not matter whether or not these are auxiliary blocks or whether you also want to give these blocks new numbers.

Copying later

The blocks to be added to the existing, changed SINAUT auxiliary blocks are adapted and any number changes to these blocks made indirectly using the symbol table of the CPU and with the aid of the 'Subscriber Administration' SINAUT configuration tool.

Step 1

Copy the rows from the symbol table of the SINAUT TD7 library with the SINAUT blocks you still require to the symbol table of the CPU. If necessary, change the numbers of these blocks in the symbol table of the CPU.

If you want to add the same blocks to further CPUs, the most efficient method is to copy the new entries from the symbol table of the first CPU to the symbol tables of the other CPUs, particularly if you have assigned different numbers to the new blocks. Changes to the numbers of these blocks are then adopted directly on the other CPUs.

Step 2

Change to the 'Subscriber Administration' SINAUT configuration tool and save again as explained in Step 3.

When the save is complete, the blocks newly added to the symbol table are stored in the block directory of the CPU, if applicable with new block numbers. They have also been recompiled and therefore adapted to their local SINAUT environment; in other words internally, they call the SINAUT auxiliary blocks under their new numbers.

Note

If a SINAUT block that you have added already exists on one of the other CPUs of the project, and if the same number changes have been made for the SINAUT auxiliary blocks, the block can also be copied from the program of the relevant CPU. In this case, however, no new entry is made in the symbol table of the target CPU as is the case when copying from the TD7 library *). You should therefore modify the symbol name manually to avoid errors during subsequent compilations.

*)This applies only to versions older than STEP 7 Version 5.1

2.1.4 Copying programs

Introduction

If several stations of a SINAUT project require the same or almost the same programme, the most efficient method used to complete the program first for one of the stations. Following this, the program can be copied to all other stations and adapted to the local situation. A safe method for copying programs in SINAUT projects is described below.

Initial situation

You have configured your SINAUT installation as already described in the section 'Configuration software for SINAUT ST7'; in other words, you have completed the connection configuration with the appropriate SINAUT configuration tool and have started to save in the 'Subscriber Administration' SINAUT tool. The result is that all the necessary SINAUT blocks already exist in all the program directories of the CPU in addition to the SDB directory and OB1. You have completed the program for one of the CPUs want to copy this to other CPUs.

Copying

With the method described below, you copy the entire content of the S7 program from the source CPU to the target CPU(s). Following this, you correct the CPU SDBs and the SINAUT data blocks (BasicData, XComData, PComData, BComData) on the target CPUs. You do this by saving in the 'Subscriber Administration' SINAUT configuration tool.

Step 1

- Open the S7 Program(..) directory on the CPU whose program you are using as a template.
- In the right-hand window, select all three subdirectories Sources, Symbols and Blocks (see figure).
- Copy the selected elements to the clipboard: Either using the 'Copy' function in the 'Edit' menu or using the Windows key combination *Ctrl + C*.

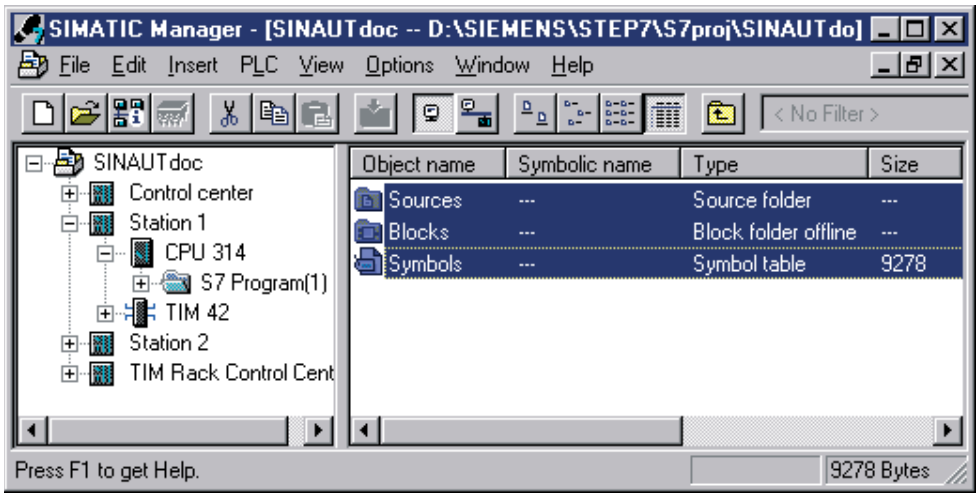


Figure 2-13 Selecting the program of the source CPU

Step 2

- Now open the S7 Program(..) directory of the CPU in which you want to insert the program from the clipboard.
- In the right-hand window, select the three subdirectories Sources, Symbols and Blocks again.
- Then start the 'Paste' function: Either using the 'Paste' function in the 'Edit' menu or using the Windows key combination *Ctrl + V*.

Repeat step 2 for all other CPUs that require the same program.

Step 3

- Go to the 'Subscriber Administration" SINAUT configuration tool and save.
- In the 'Properties' dialog box, make sure that there is a checkmark before
  - 'System data blocks for TIMS and CPUs'
    and
  - 'SINAUT TD7 blocks for CPUs'.
- Then click on 'OK'.

When the save is completed, the SINAUT-specific data has been adapted to the local situation in all CPUs and the SDBs of the CPUs once again have the correct content.

2.1.5 Using online help

Introduction

This chapter 'SINAUT TD7 software package for the CPU' is available on the PG as a help file. You can therefore call up this chapter online. The following sections describe how to call up the online help.

Starting the online help from the SINAUT TD7 library

To start the online help from the SINAUT TD7 library simply select a block in the ’Blocks’ directory, for example, FB82. Then press the F1 key. The description of the selected block is displayed immediately.

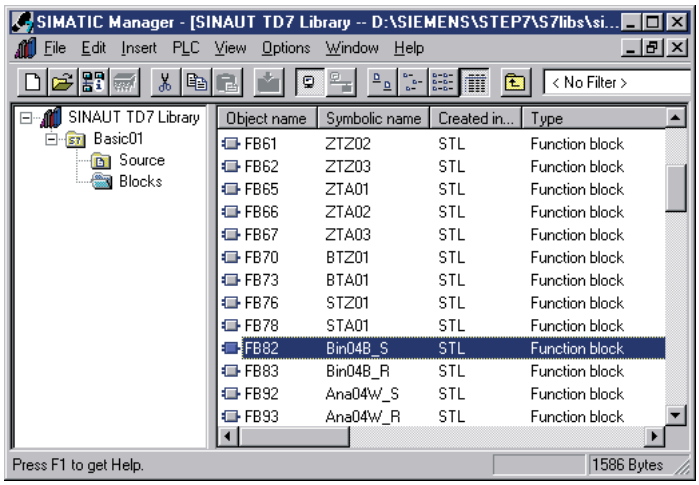

SIMATIC Manager - [SINAUT TD7 Library -- D:\SIEMENS\STEP7\S7libs\si...
File Edit Insert PLC View Options Window Help
< No Filter >
SINAUT TD7 Library
Basic01
Source
Blocks
Object name Symbolic name Created in... Type
FB61 ZTZ02 STL Function block
FB62 ZTZ03 STL Function block
FB65 ZTA01 STL Function block
FB66 ZTA02 STL Function block
FB67 ZTA03 STL Function block
FB70 BTZ01 STL Function block
FB73 BTA01 STL Function block
FB76 STZ01 STL Function block
FB78 STA01 STL Function block
FB82 Bin04B_S STL Function block
FB83 Bin04B_R STL Function block
FB92 Ana04W_S STL Function block
FB93 Ana04W_R STL Function block
Press F1 to get Help.
1586 Bytes

Figure 2-14 Starting the online help from the SINAUT TD7 library

Starting the online help from the block directory of the user program

Just as in the SINAUT TD7 library, you can start the online help from the block directory of the user program as follows: Select the required block, and then press the F1 key.

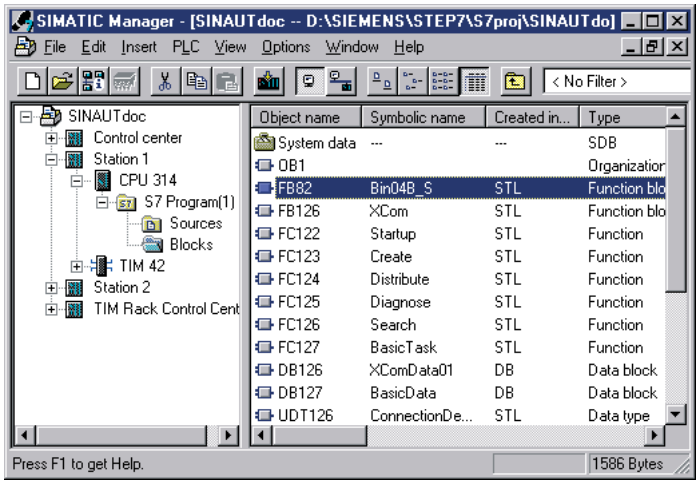


Figure 2-15 Starting the online help from the block directory of the user program

Starting online help while creating programs

You can also call up online help when you are programming. The cursor must be located in the row with the call of the block for which you need help. The figure below, for example, shows the cursor in the row with the FB82 'Bin04B_S' call. If you now press the F1 key, the help text for this block is displayed.

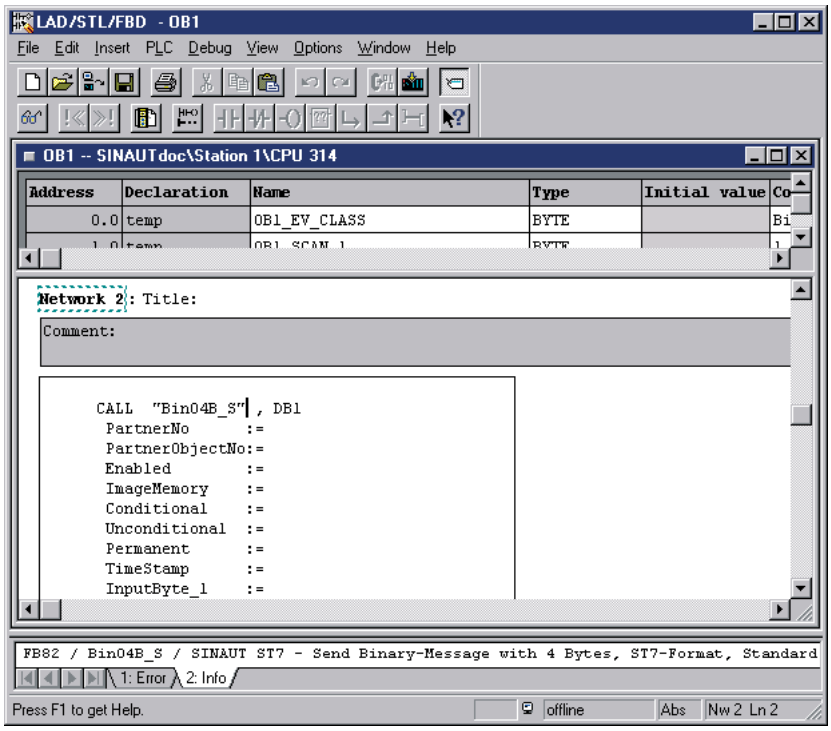


Figure 2-16 Starting online help while creating programs

Example of a help text

The following figure shows part of the help text that appears if you press the F1 key in the examples above.

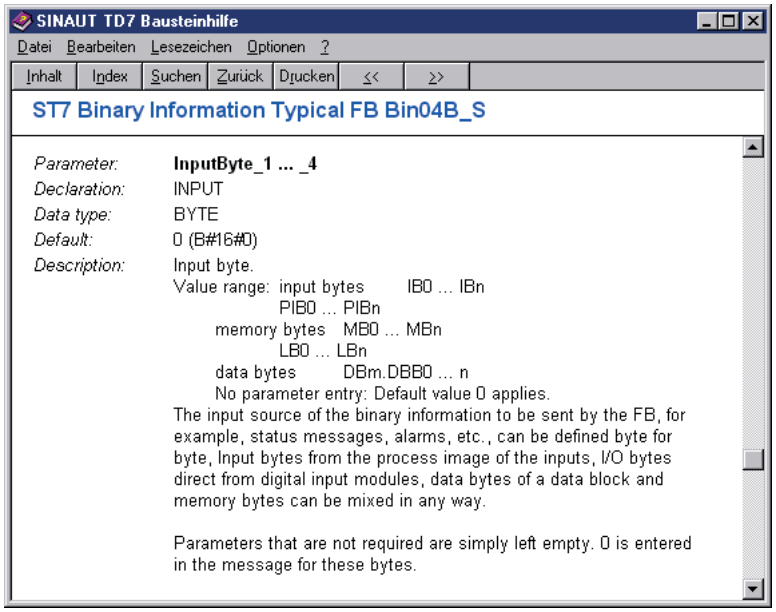

SINAUT TD7 Bausteinhilfe
Datei Bearbeiten Lesezeichen Optionen ?
Inhalt Index Suchen Zurück Drucken << >>
ST7 Binary Information Typical FB Bin04B_S
Parameter: InputByte_1 ... _4
Declaration: INPUT
Data type: BYTE
Default: 0 (B#16#0)
Description: Input byte.
Value range: input bytes IB0 ... IBn
PIB0 ... PIBn
memory bytes MB0 ... MBn
LB0 ... LBn
data bytes DBm.DBB0 ... n
No parameter entry: Default value 0 applies.
The input source of the binary information to be sent by the FB, for example, status messages, alarms, etc., can be defined byte for byte, Input bytes from the process image of the inputs, I/O bytes direct from digital input modules, data bytes of a data block and memory bytes can be mixed in any way.
Parameters that are not required are simply left empty. 0 is entered in the message for these bytes.

Figure 2-17 Example of a help text

Selecting help topics

You can access any other part of the help file from the help text for an individual block. Double-clicking on the 'Contents' button opens the help topics window: SINAUT TD7 block help' (see figure below). From here any section of this chapter 'SINAUT TD7 software package for the CPU' can be opened using the 'Contents', 'Index' or 'Find' tabs.

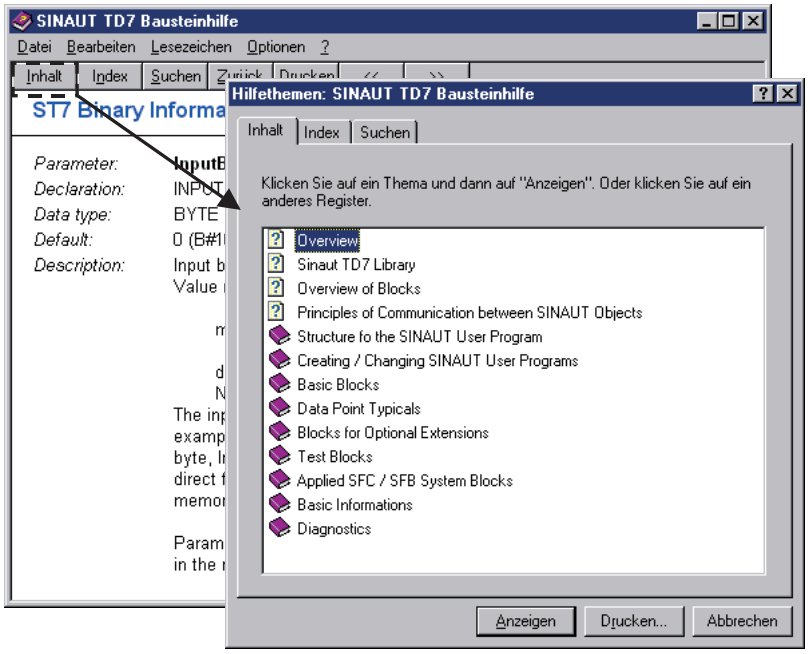


Figure 2-18 Selecting help topics

2.2 Principle of communication between SINAUT objects

Introduction

The term "object" within SINAUT ST7 refers to the type of display and handling of process entities such as status information, analog values, commands, motors, valves, regulators, etc. An object always consists of a process component and an operator control and monitoring component that operate in separate PLCs (subscribers) of the SINAUT network To perform their assigned functions, the tow components must be able to communicate with one another. Setpoints, parameters, commands and organizational instructions are sent from the operator end to the process object and the process object returns process data, alarm/status messages, and organizational information.

In the TD7 world, at both the process end as well as the operator end, an object consists of a processing instruction set - referred to below as a typical - in the form of an S7 function block (FB) and an assigned object data record in the form of an instance DB for the FB. The following figure shows the principle used for the exchange of data between the process and operator components of an object.

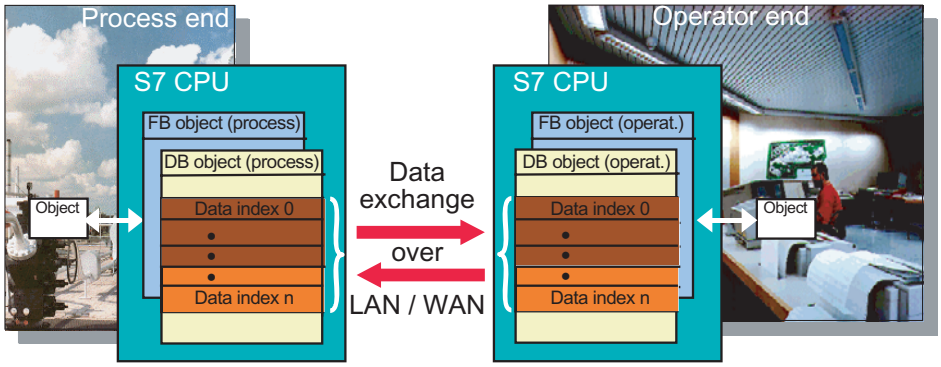


Figure 2-19 The principle of object communication

Essentially, the data that describes an object is exchanged. This is located in the object data record under data index 0...n. The extent and composition of this data area depends on the typical. It can consist of several identical data types or a combination of different data types. In the object data record at the process and operator ends, the data structure of two corresponding typicals is identical.

The data exchange does not necessarily flow in both directions. There are simple objects in which data transmission is one-way, for example, only process data is sent to the operator component.

Organizational information is also exchanged between the two ends in addition to process and operator data. This data flow and the data area reserved for this in the object data record (the org. indexes) are not shown in the figure above.

Principle of object addressing

Every automation device with SINAUT TD7 software (generally simply referred to as a PLC or CPU) is assigned a unique network-wide SINAUT subscriber number ranging from 1 to 32000. Every typical that is called in one of these CPUs has an instance DB whose number is identical to the SINAUT object number. The use of the SINAUT subscriber number and the

SINAUT object number allows unique addressing for the communication between corresponding typicals.

Each typical has the following two parameters to define the communication relation:

- PartnerNo
  Subscriber number of the partner with which data is exchanged.
- PartnerObjectNo
  Object number (= instance DB no.) on this partner.

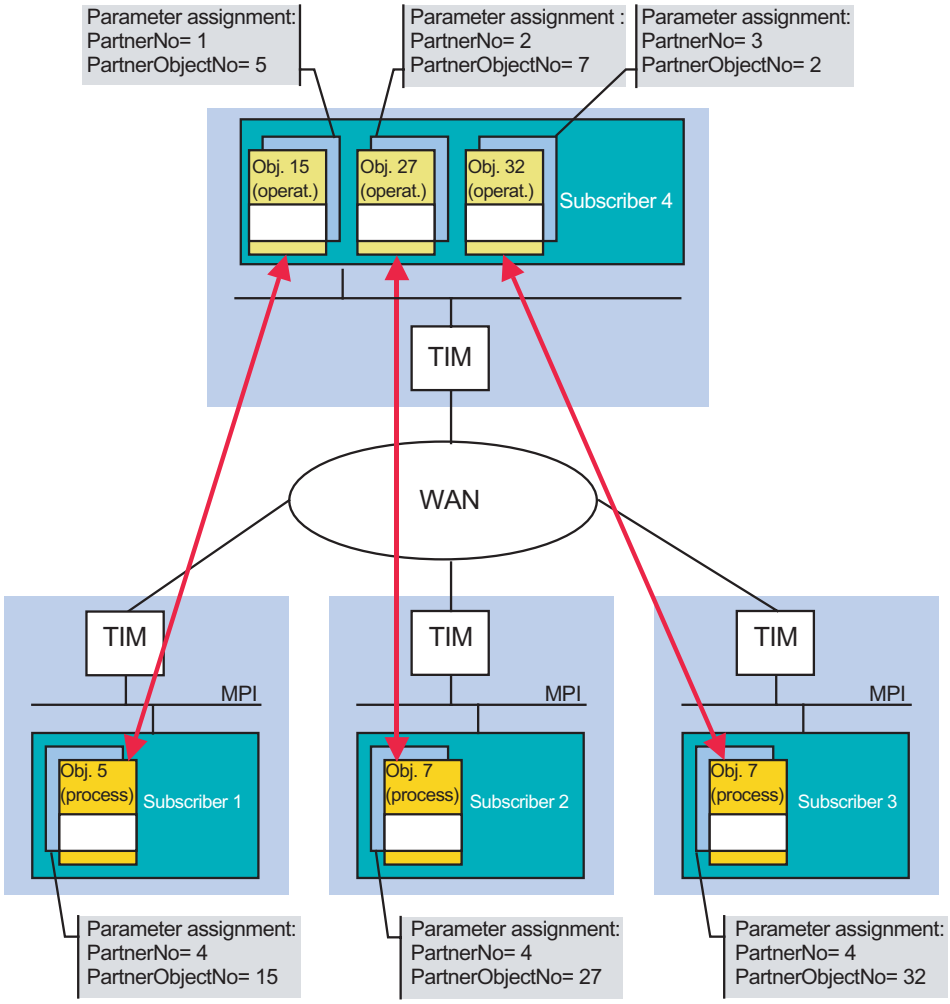


Figure 2-20 Object communication over WAN

The figure shows various examples of addressing objects that exchange data with partner objects over a WAN.

Explanation based on the example of object communication between subscribers 1 and 4:

Subscriber 1 contains the process-oriented object no. 5 and that communicates with the operator-oriented object no. 15. The partner addressing in the typical of object 5 must therefore be defined as follows

PartnerNo = 4 (subscriber number of the operator partner)

PartnerObjectNo = 15 (number of the corresponding object on this partner)

In the other direction the partner address for object 15 is as follows:

PartnerNo = 1 (subscriber number of the process partner)

PartnerObjectNo = 5 (number of the corresponding object on this partner)

When object 5 wants to transmit data to its partner object 15, a data message is put together in which the specified two-level partner address is entered as the destination address. Based on the destination subscriber 4 (= PartnerNo), the SINAUT TD7 software and the TIM modules in the SINAUT network make sure that this message is delivered to the specified destination subscriber 4.

When the message arrives at destination subscriber 4, the TD7 software reads the destination object number 15 (= PartnerObjectNo) contained in the message and recognizes that the information in the message should be stored in the local DB15. The index number contained in the message is also taken into account when selecting the storage location. If, for example, the index number is X, the information is stored in DB15 starting at data index X. Information is also entered in the object DB indicating that new data has arrived and identifying the data indexes that have been updated. In the following program cycle, the operator typical that processes this object detects the reception of new data, processes it according to the typical function and applies it to the data outputs configured for the typical.

In the opposite direction, operator object 15 sends its data intended for process object 5 in a message containing subscriber no. 1 and object no. 5 as the destination address. This message finally arrives at subscriber no. 1 directed by the TD7 software and TIMs. The information contained in the message is entered in the appropriate location in object DB 5, once again based on the destination object no. 5 and the data index. In the following program cycle, the process typical processes the newly received data and makes it available to the configured outputs in an appropriate form.

Object communication over LAN

In addition to data transmission between SINAUT objects over a WAN, the SINAUT TD7 software also allows local communication over LANs (the current software presently supports only the MPI bus as a LAN).

The following figure assumes an installation structure in which several PLCs are planned in one station. Each PLC handles a particular automation task (in the example, subscribers 10, 11 and 12). This station also includes a further higher-level PLC (subscriber 13) in which a user controller is incorporated that controls the automation in the subordinate PLCs.

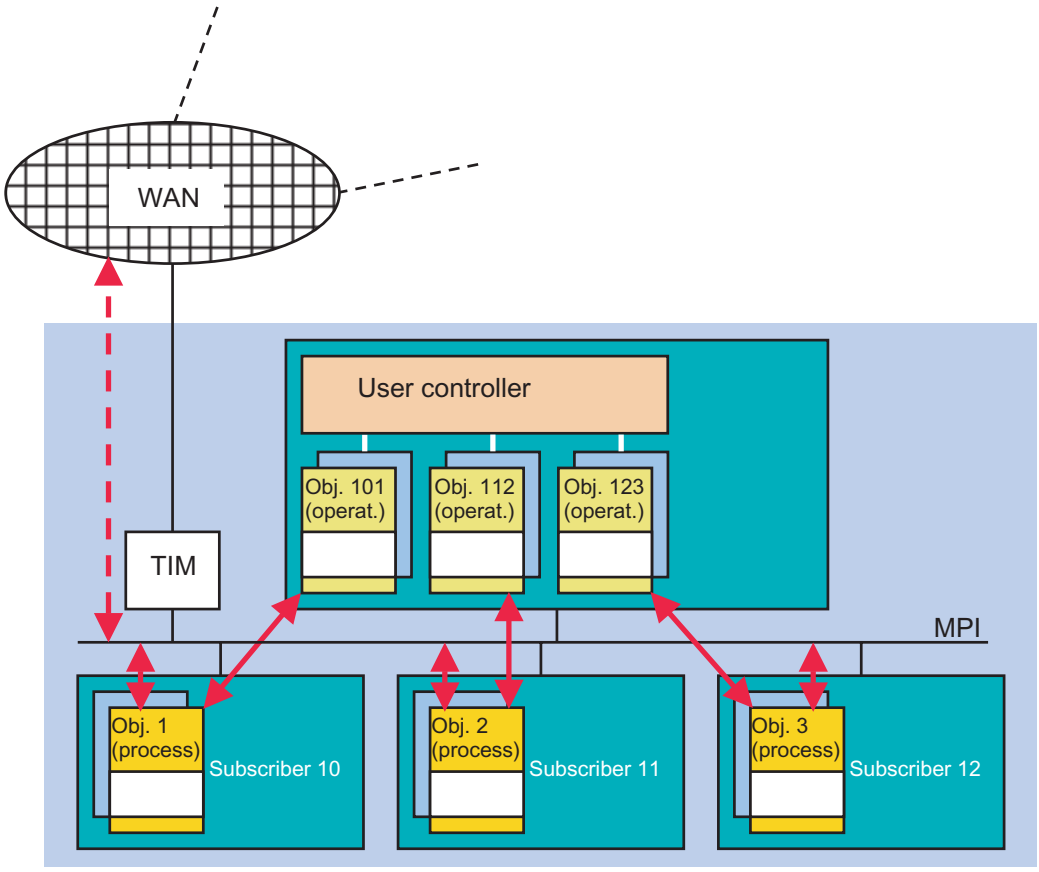


Figure 2-21 Object communication in the LAN (MPI)

In this configuration, the TD7 software could be used with its process and operator typicals as shown in the diagram. In the subordinate PLCs, the process typicals handle local automation and send data changes over the MPI bus to the operator objects in the central control device, subscriber 13. The user controller reads the process information output to the operator objects and processes it accordingly. If the current situation requires that commands or setpoints are transferred to the process objects, these are entered by the user controller over the appropriate inputs of the operator typicals. These handle the immediate transmission to the process objects that take into account this information in their automation task in keeping with the functions of the typical.

The process objects can not only send their data to their corresponding operator object in subscriber 13 and receive control instructions from there. They can also send process information to another partner, for example, a control center connected over a WAN, and, if applicable, also receive commands and setpoints from there. The following section describes how communication with more than one partner can be implemented.

Object communication with several operator subscribers

Apart from communication between one process typical and one operator typical, as in the figure above, data can also be exchanged between the process typical and more than one operator partner. The following figure illustrates the principle of such a configuration in which an operator typical for the process object in subscriber 1 is intended for local subscriber 2

and also for subscribers 3 and 4 that are connected over the WAN.

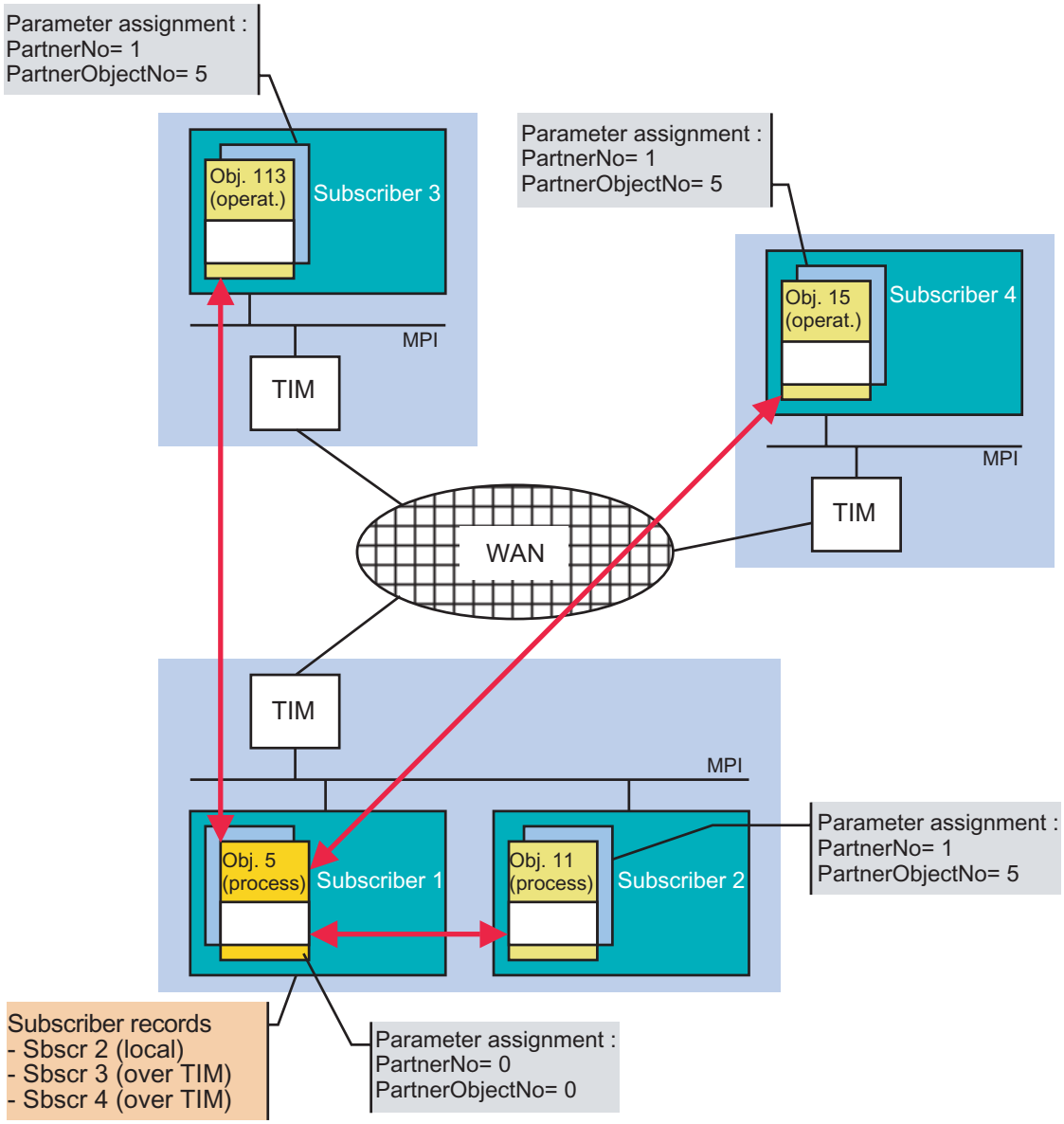


Figure 2-22 Object communication with several operator subscribers

The data acquired at the process end is sent simultaneously to all three operator partners. From the operator end, the operator data can be transferred to the process object at any time independently. Return messages from the process, resulting from an operator instruction from one subscriber, are automatically sent to all three operator objects. Each operator end therefore has the latest information. Even when it receives a general or single request, the process object sends the requested data not only to the partner who sent the request but also to the other two partner objects.

In this case, subscribers 2, 3 and 4 must either be specified explicitly as partners in process typical (the typical must then have at least three parameters for partners, which is not the

case at present) or there are no partner addresses specified in the typical. In the latter situation, the TD7 software automatically transmits to all subscribers for which a connection has been configured (using the SINAUT connection configuration). In the example above, subscriber 1 needed to have connections configured for subscriber 2 as well as for subscribers 3 and 4.

Based on this connection configuration, from the subscriber records it is known that subscriber 1 has a local connection with subscriber 2 as well as a WAN connection to subscribers 3 and 4 both over through the local TIM module. The TD7 software then puts together two messages: One message is sent to the locally available partner 2 over the MPI bus, the other message is sent to the TIM for transmission. For its part, the TIM checks which partners it knows in conjunction with subscriber 1. From its records it identifies subscriber 3 and subscriber 4.

If the WAN is a dial-up network, the TIM duplicates the message: once for subscriber 3 and once for subscriber 4. The TIM adds the destination subscriber numbers 3 or 4 to the messages that are not yet in the messages. The TIM then establishes the connection to subscriber 3 and sends the message intended for it. Following this, the TIM calls subscriber 4 and transmits its copy of the message.

If the WAN is a dedicated line network and the TIM of subscriber 3 is the polling master (master TIM), the subscriber 1 TIM cannot transmit directly to 3 and 4 as it can in a dial-up network. In this situation, the message for subscriber 4 must be transmitted indirectly over the master TIM in subscriber 3. The message to be sent is not duplicated; it is given two destination addresses, one for subscriber 3 and one for subscriber 4. This message is then to the master TIM sent with the next poll. Based on the destination addresses and its subscriber records, the TIM recognizes that one of these destinations can be reached locally and the other over the dedicated line network. The master TIM then makes a copy of the message. Destination address 4 from the original is removed and the message forwarded with the remaining destination address 3 to subscriber 3 that is available locally. The copy only includes destination address 4 and is sent to the TIM of subscriber 4 over the dedicated line from where it is then passed on to subscriber 4.

Interaction of the blocks

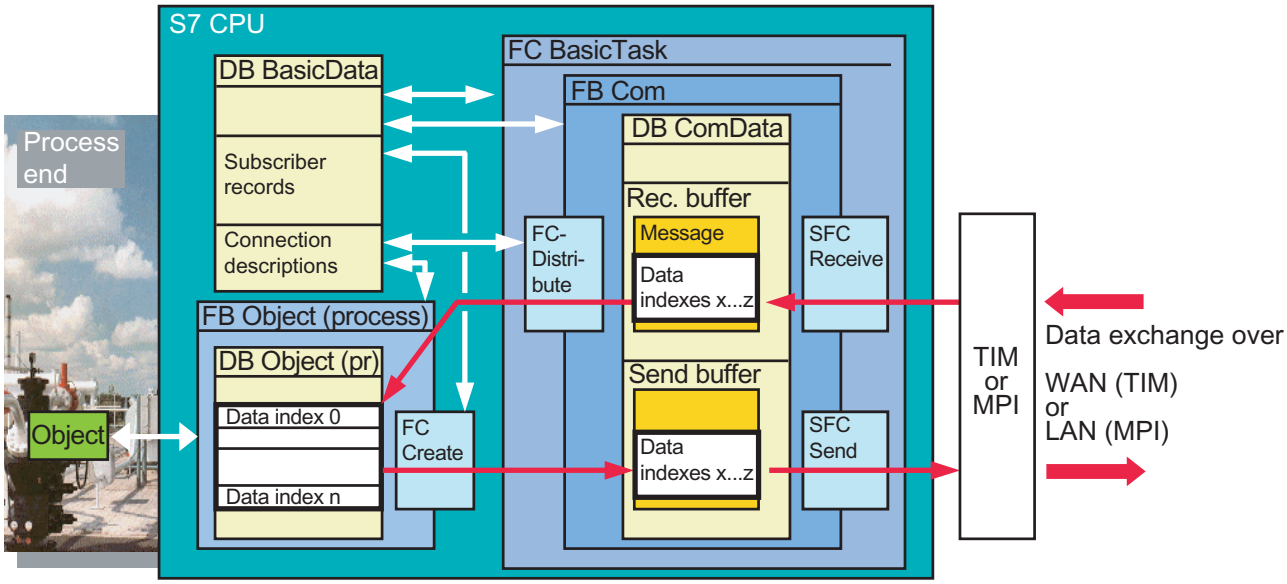


Figure 2-23 Interaction of the blocks (based on the example of the process end)

2.3 Structure of the SINAUT user program

Introduction

The SINAUT user program is contained in the following organization blocks:

- SINAUT startup program in OB100
- Cyclic SINAUT program in OB1
- Time-driven SINAUT program in a cyclic interrupt OB, for example OB35
- SINAUT test routine in the programming error OB121

Note

In the following description of the SINAUT program structure, the tasks of the individual blocks are only outlined briefly. For more detailed information on the functions and parameters, refer to the descriptions of the blocks mentioned.

2.3.1 SINAUT startup program in OB100

The structure of the SINAUT program in the startup OB100 is as follows:

Startup OB100	
Startup	• The only task required here for the SINAUT program is to integrate the FC Startup call in the startup OB100. The FC has no parameters.
	• User-specific startup functions that are required independent of the SINAUT program can be included before or after FC Startup in the startup OB.

Note

Startup OB101, which is intended for S7-400 restarts, may not be used!

2.3.2 Cyclic SINAUT program in OB1

Introduction

The basic structure of the cyclic SINAUT program in OB1 is described below. In later sections, you will find a detailed description of OB1 for a station and a master station.

Note

Unless indicated otherwise, the call sequence of the blocks must be adhered to exactly!

The entire cyclic SINAUT program must be processed in every OB1 cycle.

Basic structure of the cyclic SINAUT program in OB1

Cyclic OB1		
BasicTask	• FC BasicTask must always be called at the start of the cyclic SINAUT program. It handles basic SINAUT tasks that are always required.	
Optional SINAUT basic functions	• Directly following FC BasicTask, additional blocks can be called to perform optional basic functions, for example:	
	• - FC TimeTask	Provides the SINAUT time.
	• - FC Trigger	Scheduled starts for user programs and data messages.
	• - FC PartnerStatus	Displays subscriber OK/disrupted
	• - FC PartnerMonitor	Extended subscriber-specific display and control features.
	• - FC ListGenerator	Creation of address lists for received messages with incomplete destination addresses.
Data point typicals	• Following the FCs shown above for SINAUT basic tasks, data point typicals for sending and receiving data are called. The sequence of the individual typicals is unimportant. The number of typicals to call and the required types depend on the amount and type of data to be sent and received. You can see which data point typicals are currently available by referring to the SINAUT TD7 library installed on the programming device. All data point typicals are FBs. An instance DB must be specified when an FB is called. The number of this instance DB is identical to the object number of the datapoint object (this addressing rule does not apply to the ST1 versions of the data point typicals). • The user-specific cyclic program that is required independent of the SINAUT program can be included before or after the SINAUT program in OB1 or, if suitable, within the SINAUT program itself.	

Note

Users, of course, is free to structure the SINAUT program in OB1 according to their preferences by 'packing' the SINAUT program in one or more FCs.

The cyclic OB1 program for a station

Cyclic OB1	
BasicTask	• FC BasicTask must always be called at the start of the cyclic SINAUT program. The FC has only one parameter, namely UserFC. Normally 0 can be specified. However, if you require user-specific processing for received messages, you will need to specify the number of an FC containing the user program for this processing.
TimeTask	• As an option, you can call FC TimeTask immediately after FC BasicTask. The FC has no parameters. FC TimeTask must be included if you need the SINAUT time. This enables SINAUT messages to be time-stamped. However, you can also use the SINAUT time to start program components at a specific point in time or to schedule the transmission of data messages. FC Trigger, described below, is then required. For this FC to be used, the PLC must be provided with the SINAUT time from a local TIM module. This can be specified during the parameter assignment of the TIM in HW Config in the 'Properties' dialog, 'Time service' tab. See chapter 'Configuration software for SINAUT ST7'.
Trigger	• FC Trigger can be included as an option. The FC sets its output for the duration of one OB1 cycle when the point in time or the time interval set for the FC has been reached. The FC can be inserted several times if several times or various time intervals are required. Requirement for the use of the FC: FC TimeTask must be called first in the OB1 program (see above).
PartnerStatus	• FC PartnerStatus can be included as an option. The FC shows the current 'disrupted' or 'OK' status for a maximum of 8 SINAUT subscribers (communication partners).

Cyclic OB1	
ListGenerator	• FC ListGenerator300 (for S7-300 CPU) or FC ListGenerator400 (for S7-400 CPU) can be included as options. The FC is required if the station receives messages containing no destination address or an incomplete destination address. This can occur in the following situations: – When the station receives ST1 messages. – When the configuration of the destination address is omitted in one or more data point typicals in an ST7 device with which the station communicates (the parameters PartnerNo and PartnerObjectNo were not specified; there is therefore a transmission to all known destination subscribers). Following the FCs shown above for SINAUT basic tasks, data point typicals for sending and receiving data are called. The sequence of the individual typicals is unimportant. The number of typicals to call and the required types depend on the amount and type of data to be sent and received. The following typically applies to a station: • Send – Binary information, such as status messages and alarms – Analog values – Counted values • Receive – Commands – Setpoints (including limit values, parameters, etc.) All data point typicals are FBs. An instance DB must be specified when an FB is called. The number of this instance DB is identical to the object number of the datapoint object (this addressing rule does not apply to the ST1 versions of the data point typicals). A data point object consists of one or more data points of the same type, e.g. 4 bytes of binary information, or 4 analog values, or 1 byte commands, etc.
ST7 format	
Bin..._S	• The following should be inserted for acquiring and transmitting of binary information, such as status information, alarms, etc.: For ST7: one or more FB-Bin…_S
Ana…_S	• The following should be inserted for acquiring and transmitting analog values: For ST7: one or more FB-Ana…_S
Cnt…_S	• The following should be inserted for acquiring and transmitting counted values: For ST7: one or more FB-Cnt…_S A requirement for the use of the FBs mentioned is that FC PulseCounter is included in a cyclic interrupt OB, e.g. OB35. This FC is responsible for the actual (time-driven) acquisition of counted pulses in the background.
CMD..._R	• The following should be included for receiving and outputting commands: For ST7: one or more FB-Cmd…_R
Set..._R or Par…_R	• The following should be included for receiving and outputting setpoints, limits, parameters, etc.: For ST7: one or more FB-Set…_R or FB-Par..._R
	• The user-specific cyclic program for the station that is required independent of the SINAUT program can be included before or after the SINAUT program in OB1 or, if suitable, within the SINAUT program itself.

In the above OB1 program structure for a station, only data point typicals which process data of the same type are listed. There are additional data point typicals that can send and receive any combination of data types:

For ST7:

- FB-Dat12D_S for sending or
- FB-Dat12D_R for receiving 12 data double-words with any content

These typicals can also be included in the cyclic SINAUT program in place of or in addition to the data point typicals in the program structure here.

Note

There is no data-specific processing and change control for these typicals for any combination of information types. The user program is responsible for this. The only optional change control that can be activated is the triggering of a transmission at each bit change.

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

The cyclic OB1 program for a master station

Cyclic OB1	
BasicTask	• FC BasicTask must always be called at the start of the cyclic SINAUT program. The FC has only one parameter, namely UserFC. Normally 0 can be specified. However, if you require user-specific processing for received messages, you will need to specify the number of an FC containing the user program for this processing.
TimeTask	• As an option, you can call FC TimeTask immediately after FC BasicTask. The FC has no parameters. FC TimeTask must be included if you need the SINAUT time. This enables SINAUT messages to be time-stamped. You can also use the SINAUT time to start program sections according to a schedule. FC Trigger, described below, is then required. For this FC to be used, the PLC must be provided with the SINAUT time from a local TIM module. This can be specified during the parameter assignment of the TIM in HW Config in the 'Properties' dialog, 'Time service' tab. See chapter 'Configuration software for SINAUT ST7'.
Trigger	• FC Trigger can be included as an option. The FC sets its output for the duration of one OB1 cycle when the point in time or the time interval set for the FC has been reached. The FC can be inserted more than once if several times or various time intervals are required. Requirement for the use of the FC: FC TimeTask must be called first in the OB1 program (see above).
PartnerStatus	• FC PartnerStatus can be included as an option. The FC indicates the current 'disrupted' or 'OK' status for a maximum of 8 SINAUT subscribers (communication partners). If you want to display the status for more than 8 subscribers, a corresponding number of PartnerStatus FCs must be included. The block is practical for monitoring the connections with local TIMs.

Cyclic OB1	
PartnerMonitor	• FC PartnerMonitor can be included as an option. This FC displays important status information about a SINAUT subscriber (communication partner). The FC can also be used to trigger a general request to the subscriber and to establish and disconnect a permanent connection with the subscriber. FC PartnerMonitor must be included once for each subscriber requiring the extended display and control features. FC PartnerStatus can be omitted for these subscribers. The block is practical for monitoring and controlling the connections with ST7 stations.
ListGenerator	FC ListGenerator300 (for S7-300 CPU) or FC ListGenerator400 (for S7-400 CPU) can be included as options. The FC is required if the master station receives messages containing no destination address or an incomplete destination address. This can occur in the following situations: • When the configuration of the destination address is omitted in one or more data point typicals in an ST7 station (the parameters PartnerNo and PartnerObjectNo were not specified; there is therefore a transmission to all known destination subscribers). • When ST1 stations are connected to the master station. Following the FCs shown above for SINAUT basic tasks, data point typicals for sending and receiving data are called. The sequence of the individual typicals is unimportant. The number of typicals to call and the required types depend on the amount and type of data to be sent and received. The following typically applies to master station: • Send – Commands – Setpoints (including limit values, parameters, etc.) • Receive – Binary information, such as status messages and alarms – Analog values – Counted values All data point typicals are FBs. An instance DB must be specified when an FB is called. The number of this instance DB is identical to the object number of the datapoint object (this addressing rule does not apply to the ST1 versions of the data point typicals). A data point object consists of one or more data points of the same type, e.g. 4 bytes of binary information, or 4 analog values, or 1 byte commands, etc.
ST7 format	
Bin..._R	• The following should be inserted for receiving and outputting binary information, such as status information, alarms, etc.: For ST7: one or more FB-Bin…_R
Ana…_R	• The following should be included for receiving and outputting analog values: For ST7: one or more FB-Ana…_R
Cnt…_R	• The following should be included for receiving and outputting counted values: For ST7: one or more FB-Cnt…_R

Cyclic OB1	
CMD..._S	• The following should be inserted for acquiring and transmitting commands: For ST7: one or more FB-Cmd…_S A requirement for the use of the FBs mentioned is that FC Safe is included at the end of all data point typicals. This FC is responsible for reliable input of commands and setpoints.
Set..._S **or** **Par…_S**	• The following should be included for acquiring and transmitting setpoints, limits, parameters, etc.: For ST7: one or more FB-Set…_S or FB-Par..._S A requirement for the use of the FBs mentioned is that FC Safe is included at the end of all data point typicals. This FC is responsible for reliable input of commands and setpoints.
Safe	• FC Safe must be called once at the end of all data point typicals when send blocks for commands (Cmd..._S, BTZ..) or setpoints (Set..._S, Par..._S, STA..) are called. • The user-specific cyclic program for the master station that is required independent of the SINAUT program can be included before or after the SINAUT program in OB1 or, if suitable, within the SINAUT program itself.

In the above OB1 program structure for a master station, only data point typicals which process data of the same type are listed. There are additional data point typicals that can send and receive any combination of data types:

For ST7:

- FB-Dat12D_S for sending or
- FB-Dat12D_R for receiving 12 data double-words with any content

These typicals can also be included in the cyclic SINAUT program in place of or in addition to the data point typicals in the program structure here.

Note

There is no data-specific processing and change control for these typicals for any combination of information types. The user program is responsible for this. The only optional change control that can be activated is the triggering of a transmission at each bit change.

Note

In the master station it is practical to structure the OB1 program according to stations, in other words, all send and receive data typicals belonging to the same station are packaged in one FC. The best overview is provided when the number of the FC is identical to the subscriber number of the station.

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

2.3.3 Time-driven SINAUT program in a cyclic interrupt OB

Introduction

A time-driven SINAUT program is only needed in a CPU if counted pulse acquisition is used in the CPU.

The counted pulses are acquired over a normal digital input module. To be able to acquire the pulses reliably, the digital inputs used must be polled for changes at fixed time intervals. This time interval is based on the duration of the shortest counted pulse. The minimum permitted counted pulse duration is 50 ms. The same applies to the length of the pause. The resulting maximum count frequency is 10 Hz.

The time interval at which the counted pulse should be acquired must be approximately half of the counted pulse duration, in other words, an interval of approx. 25 ms is used for a 50 ms pulse.

For this time-driven counted pulse acquisition, OB35 must be programmed for an S7-300 CPU and one of the available cyclic interrupt OBs, OB30 to OB38, for an S7-400 CPU. Although all of the cyclic interrupt OBs have a preset time interval (for example, 100 ms for OB35), this can be changed in 1 ms steps. For example, a cyclic interrupt OB can be set to 25 ms.

The figure below shows how to change the time interval for a cyclic interrupt OB in the HW Config Properties dialog for the CPU.

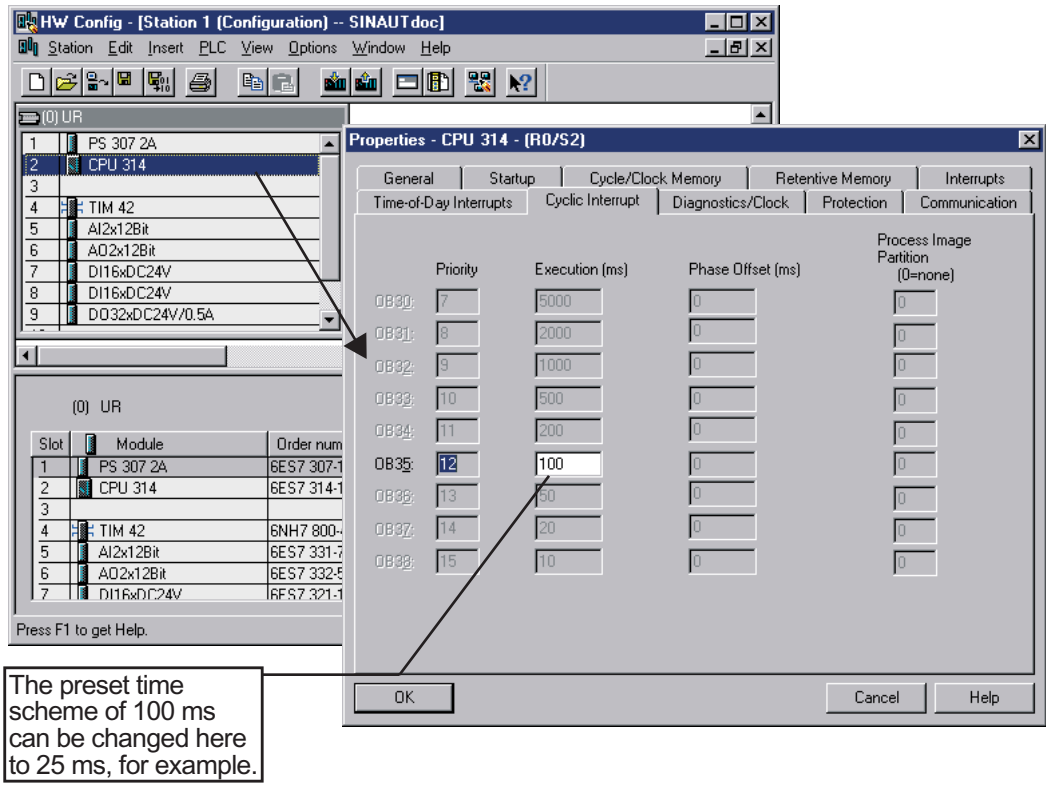

HW Config - [Station 1 (Configuration) -- SINAUTdoc]
Station Edit Insert PLC View Options Window Help
(0) UR
1 PS 307 2A
2 CPU 314
3
4 TIM 42
5 AI2x12Bit
6 AO2x12Bit
7 DI16xDC24V
8 DI16xDC24V
9 DO32xDC24V/0.5A
(0) UR
Slot Module Order num
1 PS 307 2A 6ES7 307-1
2 CPU 314 6ES7 314-1
3
4 TIM 42 6NH7 800-
5 AI2x12Bit 6ES7 331-7
6 AO2x12Bit 6ES7 332-5
7 DI16xDC24V 6ES7 321-1
Press F1 to get Help.
Properties - CPU 314 - (R0/S2)
General Startup Cycle/Clock Memory Retentive Memory Interrupts
Time-of-Day Interrupts Cyclic Interrupt Diagnostics/Clock Protection Communication
Priority
Execution (ms)
Phase Offset (ms)
Process Image Partition (0=none)
OB30: 7 5000 0 0
OB31: 8 2000 0 0
OB32: 9 1000 0 0
OB33: 10 500 0 0
OB34: 11 200 0 0
OB35: 12 100 0 0
OB36: 13 50 0 0
OB37: 14 20 0 0
OB38: 15 10 0 0
OK
Cancel
Help
The preset time scheme of 100 ms can be changed here to 25 ms, for example.

Figure 2-24 Changing the time interval for cyclic interrupt OB35

The program structure in a cyclic interrupt OB

The structure of the SINAUT program in the cyclic interrupt OB is as follows:

Cyclic interrupt OB	
PulseCounter	• One or more PulseCounter FCs can be inserted for acquisition of counted pulses. FC PulseCounter works with up to 8 pulse inputs from any type of digital input. The acquired counted pulses are accumulated in configurable SIMATIC counters that are available to the function blocks that put together the counted value messages (FB Cnt..._S, FB ZTZ..). For more detailed information, refer to the description of the SINAUT program structure in OB1. • User-specific cyclic interrupt functions that are required independent of the SINAUT program, can be inserted at any location in the cyclic interrupt OB.

2.3.4 SINAUT test routine in the programming error OB121

Introduction

When a non-existent block is called in a CPU, the CPU usually changes to STOP and the missing block (FBxx, FCyy or DBzz) is indicated in the diagnostics buffer. You can then load the missing block and restart the CPU. If you want to avoid the CPU stopping when there is a missing block, or only changing to STOP when certain block types or block numbers are missing, you can specify the type of response you require in OB121 with a user program. The CPU will continue to operate despite a missing block simply by loading OB121 even it contains nothing. If you want to specify when the CPU should continue and when it should stop, you will need to include an appropriate user program in OB121.

In conjunction with SINAUT ST7, it is possible that a CPU will stop if it receives a message that it does not know (or does not yet know) from another CPU. For example, when you add a datapoint typical to a station and assign it a complete destination address (destination subscriber number plus destination object number). The set destination object number may result in a stop on the destination subscriber because the message is transmitted to the destination as soon as the new datapoint typical is installed in the station. If, however, the corresponding receive typical has not yet been installed in the destination CPU, the destination object number (instance DB of the receive typical to be installed) is also not yet available. The result is that the CPU stops as soon as this message is received, unless you have configured OB121 to avoid this. For SINAUT ST7 CPUs it is recommended that you call the FC ST7ObjectTest function in OB121. Then the CPU does not stop when a SINAUT object DB is not available as in the example described.

FC ST7ObjectTest has a StopInOtherCases parameter. It allows you to specify what should occur in other situations: Stop or continue operation when OB121 is called because another data block, an FB or an FC is missing.

The program structure in the programming error OB121

The structure of the SINAUT program in the programming error OB121 is as follows:

OB121	
ST7ObjectTest	• Calling FC ST7ObjectTest in OB121 prevents a CPU stop when the CPU receives a message with an unknown destination object number. FC-ST7ObjectTest has a StopInOtherCases parameter. It allows you to specify what should occur in other situations: Stop or continue operation when OB121 is called because another data block, an FB or an FC is missing. • A user-specific cyclic program that is required independent of the SINAUT program can be included at any location in OB121.

2.4 Basic blocks

These blocks are needed for organizational tasks within a CPU and for controlling and monitoring all the transmission channels.

2.4.1 FC Startup

This block is required in every CPU. It must be included in the startup program OB100. Its only task is to set the startup memory bit in the DB BasicData and reset the corresponding edge memory bit if it is still set.

The block has no parameters.

In a normally configured SINAUT installation, FC Startup is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

For more detailed information on the SINAUT startup program, refer to the section 'SINAUT startup program in OB100’.

2.4.2 FC BasicTask

Function

This block is required in every CPU. It handles

- The central tasks to be performed during startup
- The processing of all communication channels
- The central organizational tasks such as starting and monitoring general requests, responding to general requests, etc.

In a normally configured SINAUT installation, FC BasicTask is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration

tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

Note

The FC BasicTask must always be called as the first block in the cyclic SINAUT program (in OB1).

Explanation of the parameters

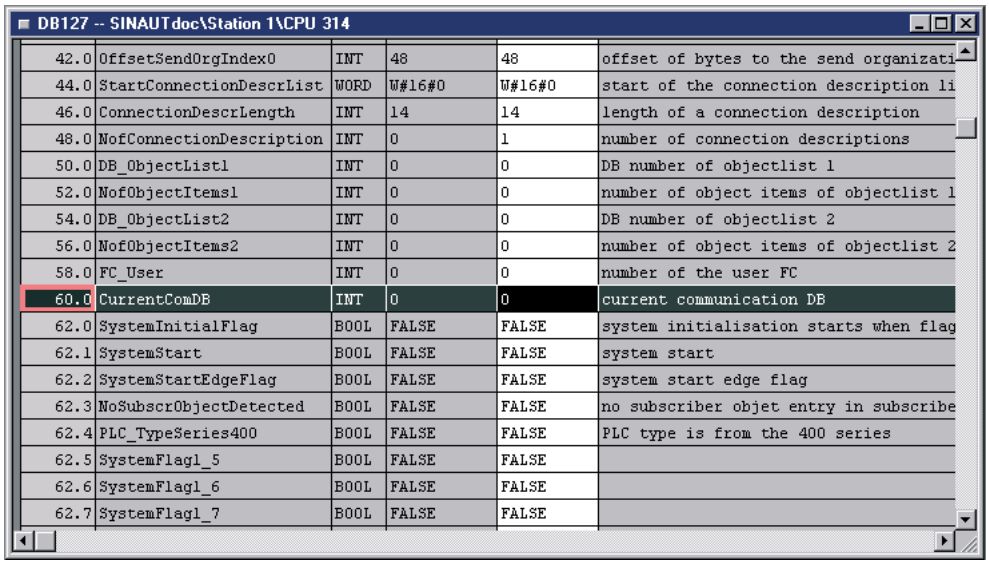
Name:	**UserFC**		
Declaration:	INPUT		
Data type:	INT		
Explanation:	Number of a user FC for user-specific processing of received SINAUT messages.		
	Range of values:	0 or 1 ... 32000	
		0 =	Dummy value in case no user FC is available for the specified purpose.
		1 ... 32000 =	Number of the user FC. The maximum number possible depends on the CPU where the SINAUT program will run.

If a user FC is specified, this FC will be called automatically by the SINAUT program each time a message is received. At the time of the call, the received message is still in the receive buffer of the communication DB. The user program in the user FC can read the received message from the receive buffer and process it as necessary, for example, transfer it to a temporary buffer.

The number of the current communication DB can be read by the user program from DW60 of DB BasicData (symbolic address: CurrentComDB).

As soon as the communications DB is opened, the beginning of the receive message in the receive buffer can be found with the pointer in DW10 (symbolic address: CurrentReceivedMessage).

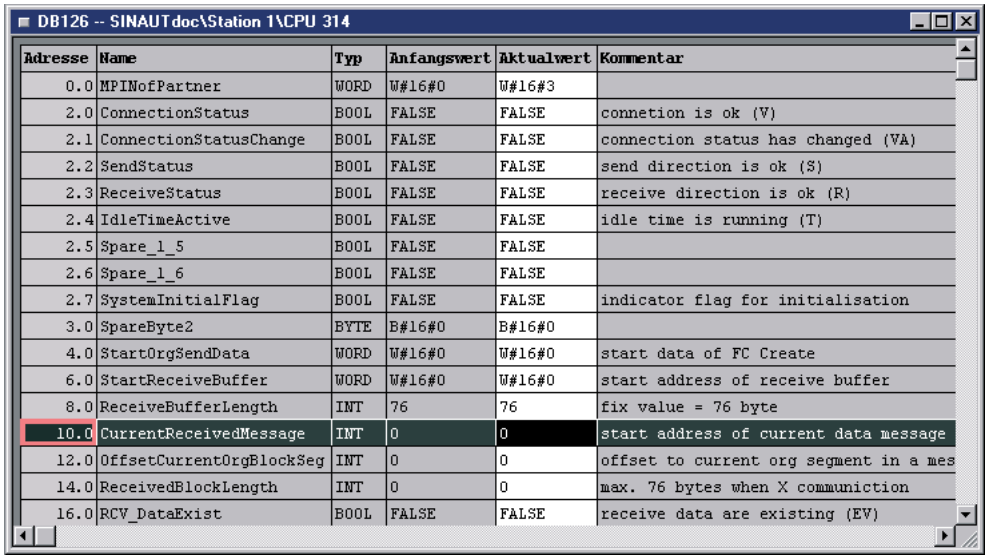
There is a separate description of the structure of a message available in the receive buffer.



DB127 -- SINAUTdoc\Station 1\CPU 314

42.0	OffsetSendOrgIndex0	INT	48	48	offset of bytes to the send organizati
44.0	StartConnectionDescrList	WORD	W#16#0	W#16#0	start of the connection description li
46.0	ConnectionDescrLength	INT	14	14	length of a connection description
48.0	NofConnectionDescription	INT	0	1	number of connection descriptions
50.0	DB_ObjectList1	INT	0	0	DB number of objectlist 1
52.0	NofObjectItems1	INT	0	0	number of object items of objectlist 1
54.0	DB_ObjectList2	INT	0	0	DB number of objectlist 2
56.0	NofObjectItems2	INT	0	0	number of object items of objectlist 2
58.0	FC_User	INT	0	0	number of the user FC
60.0	CurrentComDB	INT	0	0	current communication DB
62.0	SystemInitialFlag	BOOL	FALSE	FALSE	system initialisation starts when flag
62.1	SystemStart	BOOL	FALSE	FALSE	system start
62.2	SystemStartEdgeFlag	BOOL	FALSE	FALSE	system start edge flag
62.3	NoSubscrObjectDetected	BOOL	FALSE	FALSE	no subscriber objet entry in subscribe
62.4	PLC_TypeSeries400	BOOL	FALSE	FALSE	PLC type is from the 400 series
62.5	SystemFlag1_5	BOOL	FALSE	FALSE	
62.6	SystemFlag1_6	BOOL	FALSE	FALSE	
62.7	SystemFlag1_7	BOOL	FALSE	FALSE	

Figure 2-25 DB-BasicData, DW60 CurrentComDB Number of the current communication DB



DB126 -- SINAUTdoc\Station 1\CPU 314

Adresse	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	MPINofPartner	WORD	W#16#0	W#16#3	
2.0	ConnectionStatus	BOOL	FALSE	FALSE	connetion is ok (V)
2.1	ConnectionStatusChange	BOOL	FALSE	FALSE	connection status has changed (VA)
2.2	SendStatus	BOOL	FALSE	FALSE	send direction is ok (S)
2.3	ReceiveStatus	BOOL	FALSE	FALSE	receive direction is ok (R)
2.4	IdleTimeActive	BOOL	FALSE	FALSE	idle time is running (T)
2.5	Spare_1_5	BOOL	FALSE	FALSE	
2.6	Spare_1_6	BOOL	FALSE	FALSE	
2.7	SystemInitialFlag	BOOL	FALSE	FALSE	indicator flag for initialisation
3.0	SpareByte2	BYTE	B#16#0	B#16#0	
4.0	StartOrgSendData	WORD	W#16#0	W#16#0	start data of FC Create
6.0	StartReceiveBuffer	WORD	W#16#0	W#16#0	start address of receive buffer
8.0	ReceiveBufferLength	INT	76	76	fix value = 76 byte
10.0	CurrentReceivedMessage	INT	0	0	start address of current data message
12.0	OffsetCurrentOrgBlockSeg	INT	0	0	offset to current org segment in a mes
14.0	ReceivedBlockLength	INT	0	0	max. 76 bytes when X communiction
16.0	RCV_DataExist	BOOL	FALSE	FALSE	receive data are existing (EV)

Figure 2-26 Current communication DB, DW10 CurrentReceivedMessage,
Pointer to the start of the current received message in the receive buffer

2.4.3 DB BasicData

This data block handles the central data storage; in other words, information that needs to be kept at a central location for all blocks. Among other things, the data block includes the subscriber records and the connection descriptions.

DB BasicData is automatically created with the necessary length, assigned the data specific to the subscribers and connections and then saved in the block directory of the CPU. This

happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

DB BasicData replaces, among other things, the data blocks known from the TD1 package, ’stations record’, ’control center record’, ’interfaces record’ and the TD1 system bit memory address area from MB142 to MB199.

DB BasicData is available once on every CPU.

Note

In the SINAUT TD7 library, DB BasicData has the number DB127 and is also saved under this number in the CPUs when it is created for them. In principle, it would be possible to change the number but this requires a great deal of effort and may lead to errors when creating the rest of the SINAUT program. We therefore recommend: Leave DB127 free for DB BasicData if at all possible!

2.4.4 FB XCom

Auxiliary block for FC BasicTask, for processing a communication buffer of type DB XComData, in which a unconfigured connection (X connection) is handled using the SFCs X_SEND and X_RCV.

FB XCom also ensures that received messages are distributed immediately to the corresponding receive objects in the CPU. To do this, FB XCom calls FC Distribute as an auxiliary block.

In a normally configured SINAUT installation, FC XCom is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active. During compilation, there is a check to establish the CPU for which the program is being compiled. If an S7-300 CPU is involved, the FB-XCom is entered in the block directory of the CPU. With an S7-400 CPU, this is FB-BCom.

2.4.5 DB XComData

Instance data block for the communication block FB XCom. This communication DB contains a receive and a send buffer as well as central data required for the control and management of the X connection handled by this DB.

The data block is required in every CPU in which FB XCom is used. It can be inserted several times if the CPU maintains several X connections.

DB XComData is automatically created with the necessary length, assigned the data specific to the subscribers and connections and then saved in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

2.4.6 FB-PCom

Auxiliary block for FC BasicTask, for processing a communication buffer of type DB-PComData using SFCs WR_REC and RD_REC. Received messages are then distributed immediately to the relevant receive objects in the CPU. This is achieved by FB-PCom calling FC Distribute as an auxiliary block.
The FB-PCom block is used only with normal communication over the P bus. This relates to communication between a TIM module and a CPU module of the type series CPU 317 and CPU 319 and the CPU 315-2PN/DP.

2.4.7 DB PComData

Instance data block for the communication block FB PCom. This communication DB contains a receive and a send buffer as well as central data required for the control and management of the connection handled by this buffer.
The data block is required in every CPU in which FB PCom is used. It can be inserted several times if the CPU maintains several such connections.

2.4.8 FB BCom

Auxiliary block for FC BasicTask, for processing a communication buffer of type DB BComData, in which a configured connection (communication block connection) is handled using the SFBs BSEND and BRCV.

FB BCom also ensures that received messages are distributed immediately to the corresponding receive objects in the CPU. To do this, FB BCom calls FC Distribute as an auxiliary block.

In a normally configured SINAUT installation, FC BCom is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active. During compilation, there is a check to establish the CPU for which the program is being compiled. If an S7-400 CPU is involved, the FB BCom is entered in the block directory of the CPU. With an S7-300 CPU, this is FB XCom.

2.4.9 DB BComData

Instance data block for the communication block FB BCom. This communication DB contains a receive and a send buffer as well as central data required for the control and management of the communication block connection handled by this DB.

The data block is required in every CPU in which FB BCom is used. It can be inserted several times if the CPU maintains several communication block connections.

DB BComData is automatically created with the necessary length, assigned the data specific to the subscribers and connections and then saved in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

2.4.10 FC Create

Auxiliary block for putting together messages and entering them in one or more relevant send buffers. This block is required by the data point typicals for data and organizational messages and by FC BasicTask only for organizational messages.

This block is required in every CPU.

In a normally configured SINAUT installation, FC Create is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

2.4.11 FC Distribute

Auxiliary block for distributing messages from the receive buffer to the appropriate data point typicals or to the subscriber objects in the subscriber records.

This block is required in every CPU.

In a normally configured SINAUT installation, FC Distribute is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

2.4.12 FC Search

Auxiliary block for searching:

- For the initial address of a subscriber object within the subscriber records
- For the local object no. (instance DB) from one of the two object reference lists for a received message with an incomplete destination address

This auxiliary block is required by almost all blocks. This block is required in every CPU.

In a normally configured SINAUT installation, FC Search is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

2.4.13 FC Diagnose

Auxiliary block for entering SINAUT system messages in the diagnostic buffer of the CPU.

This block is required in every CPU.

In a normally configured SINAUT installation, FC Diagnose is automatically available in the block directory of the CPU. This happens as soon as you save in the SINAUT configuration tool ’Subscriber Administration’ and the option for compiling the 'SINAUT TD7 Blocks for the CPUs' is marked as active.

2.5 Data point typicals

Introduction

Data point typicals consist of one or more data points of the same type, e.g. 4 bytes of binary information, or 4 analog values, or 1 byte commands, etc.

Data point typicals for a specific type or amount of information always come in two versions:

- A typical for acquiring and sending
  For example, FB Bin04B_S, for acquiring and sending 4 bytes of binary information (status signals);
- A typical for receiving and outputting
  For example, FB Bin04B_R, for receiving and outputting 4 bytes of binary information (status signals).

All data point typicals are FBs. An instance DB must be specified when an FB is called. The number of this instance DB is identical to the object number of the data point object. At the send and receive ends this object number does not have to be identical.

Structure of typical names

Data point typicals have 8-character names based on the following scheme:

Table 2- 3 Structure of 8-character typical names

<table>
<tr><th colspan="3">1</th><th>2</th><th>3</th><th colspan="2">4</th><th>5</th><th>6</th><th>7</th><th>8</th></tr>
<tr><td colspan="3">Column 1 ... 3
Data point type:</td><td colspan="2" rowspan="8">Column 4 ... 5
Amount of data in the format of column 6.</td><td colspan="2" rowspan="8">Data format:
X = bit
B = byte
W = word
D = double word
R = real</td><td rowspan="8">Not used
(filled by underscore)</td><td rowspan="8">S = send function

R = receive function</td></tr>
<tr><td>Bin</td><td colspan="2">Binary information (status signal, alarm)</td></tr>
<tr><td>Ana</td><td colspan="2">Analog value</td></tr>
<tr><td>Cnt</td><td colspan="2">Counted value</td></tr>
<tr><td>Cmd</td><td colspan="2">Command</td></tr>
<tr><td>Set</td><td colspan="2">Setpoint (setpoint, parameter)</td></tr>
<tr><td>Par</td><td colspan="2">Parameter</td></tr>
<tr><td>Dat</td><td colspan="2">Data (any mixture of information types)</td></tr>
</table>

The table above relates to the name structure for ST7 typicals.

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

Overview of the available data point typicals

The following table provides an overview of the currently available data point typicals for data exchange between two SINAUT ST7 subscribers.

Table 2- 4 Overview of the available data point typicals

SINAUT protocol	Symbolic FB name	Explanation
Message typicals		
ST7	Bin04B_S	ST7 status message object, send 4 bytes of status/binary information
	Bin04B_R	ST7 status message object, receive 4 bytes of status/binary information
Analog value typicals		
ST7	Ana04W_S	ST7 analog value object, send 4 analog values (16-bit value in the INT format)
	Ana04W_R	ST7 analog value object, receive 4 analog values (16-bit value in the INT format)
Counted value typicals		
ST7	Cnt01D_S	ST7 counted value object, send 1 counted value (32-bit ST1 format)
	Cnt01D_R	ST7 counted value object, receive 1 counted value (32-bit ST1 format)
	Cnt04D_S	ST7 counted value object, send 4 counted values (32-bit ST1 format)
	Cnt04D_R	ST7 counted value object, receive 4 counted values (32-bit ST1 format)
Command typicals		
ST7	Cmd01B_S	ST7 command object, send 1 byte commands (1-out-of-8 ST1 format)
	Cmd01B_R	ST7 command object, receive 1 byte commands (1-out-of-8 ST1 format)
Setpoint/parameter typicals		
ST7	Set01W_S	ST7 setpoint object, receive 1 setpoint (16 bits) and receive current local setpoint
	Set01W_R	ST7 setpoint object, receive 1 setpoint (16 bits) and send current local setpoint
	Par12D_S	ST7 parameter object, send max. 12 double words with parameters and receive current local parameters.
	Par12D_R	ST7 parameter object, receive max. 12 double words with parameters and send current local parameters.
Other data typicals		
ST7	Dat12D_S	ST7 data object, send max. 12 double words with any information.
	Dat12D_R	ST7 data object, receive max. 12 double words with any information.

Notes on the SINAUT Time stamp

For many data point typicals you can use the TimeStamp parameter to instruct that the data object should be transferred with a time stamp. However for the data point typicals used to receive this data there is no output parameter with which to output the received time stamp. The time stamp is only saved in the instance DB which you have specified when calling the respective receive typical. This occurs in two data double words that always have the same name in all instance DBs (SINAUT object DB), namely:

Name of the double word	Contents
RecTimeStamp_1	Year, month, day and hour
RecTimeStamp_2	Minute, second, millisecond and time status

The date and time of day is coded in BCD format (exception: the half byte with the time status). The exact byte-for-byte content appears as follows:

Name of the double word	Byte no.	Contents	
		High nibble	Low nibble
`RecTimeStamp_1`	0	Year * 10	Year * 1
	1	Month * 10	Month * 1
	2	Day * 10	Day * 1
	3	Hour * 10	Hour * 1
`RecTimeStamp_2`	0	Minute * 10	Minute * 1
	1	Second * 10	Second * 1
	2	Millisecond * 100	Millisecond * 10
	3	Millisecond * 1	Time status

The content of the half byte with the time status bit::

Bit place	Value	Meaning
2^0	0	Time is invalid
	1	Time is valid
2^1	0	Standard time
	1	Daylight saving time
2^2		Not used
2^3		Not used

The time double words occupy different addresses depending on the typical. Look in the instance DB or in the declaration header of the FB to find the absolute address of both double words. It is more convenient to give the instance DBs symbolic names. You can then use the symbolic addresses to read out the information. In this case, you do not need to worry about the actual absolute addresses. These are used automatically by STEP 7. The following example clarifies this procedure.

Example

Symbolic name of instance DB: `ObjectDB27`

The STEP 7 program for reading the date and time of day and for saving in DB20 beginning with data byte 100 may appear as follows programmed in STL:

```
L"ObjectDB27".RecTimeStamp_1
TDB 20.DBD 100
L"ObjectDB27".RecTimeStamp_2
TDB 20.DBD 104
```

Notes on *explanation of the parameters*

The detailed descriptions of the blocks in most of the following contain an *"Explanation of the parameters"* section. The following information is available there for each parameter:

Name:	**Name of the parameter**
Declaration:	SIMATIC parameter type INPUT, OUTPUT or IN_OUT
Data type:	SIMATIC data type
	The data types used:
	BOOL, BYTE, WORD, DWORD, INT, DINT, ANY, COUNTER and BLOCK_DB
Default	(only applicable for FB parameters)
	Default value for the parameter. This value is valid when the parameter is not specified when the FB is called.
Explanation:	Detailed description of the parameter and specification of the allowable value range.

Note

The data point typicals are described in detail in the following pages. Data point typicals that are identical except for the number of data points to be processed are included in a single description.

If there is a difference in the function or in the parameters, the block for which the described function or parameter is valid is shown in square brackets, for example, [Cnt01D_S].

2.5.1 Parameters you require often

Parameters you require often

Before coming to the others, the next section describes commonly recurring parameters used by several of the data point typicals.

PartnerNo

Name:	**PartnerNo**
Declaration:	INPUT
Data type	INT
Default:	0
Explanation	Subscriber no. of the partner

Range of values: 0 or 1 ... 32000

The subscriber number of the partner with which the FB communicates, i.e. to which the FB sends data or from which the FB receives data must be specified.

For a process typical, this is usually the subscriber no. of the master PLC or the ST7cc control center.

With an operator typical, this is normally the subscriber number of a station PLC.

Points to note when PartnerNo = 0 in various typical classes:

- **Sending process typicals**
  (Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Dat12D_S)

  When the parameter is set to 0, the data is transmitted to all subscribers for which a connection has been configured. The following PartnerObjectNo parameter is then irrelevant.

  If the set PartnerNo was not found in the administration (in DB-BasicData), an entry to this effect is made in the diagnostic buffer (event ID B101). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

  Caution
  If "PartnerNo = 0" is set, make sure that each partner sends the message with a complete destination address (target subscriber no. and target object no.).

  Note
  When using the block in the PLC of a node station, you should consider the consequences of PartnerNo = 0 If the PLC of the node station maintains both connections to higher-level subscribers as well as to lower-level stations, a message with PartnerNo = 0 is transferred to all subscribers both "up" and "down".

- **Receiving process typicals**
  (Cmd01B_R, Set01W_R, Par12D_R)

  Set the parameter to 0 when the typical can receive data from more than one partner, for example, when there are several control centers wanting to send data to the typical configured here.

  If the PLC receives a message for the object set here, and PartnerNo is greater than 0, the system checks whether the source subscriber number in the message is identical to the PartnerNo set here. If they are different, the received information is discarded. An error message to this effect is entered in the diagnostic buffer (event ID B130).

  This check is not made if PartnerNo = 0. Regardless of the sender, each message addressed to the object is also passed on to the object.

  If the set PartnerNo is greater than 0 and this number was not found in the administration (in DB-BasicData), an entry is made in the diagnostic buffer (event ID B101). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

  Caution
  If "PartnerNo = 0" is set, make sure that each partner sends the message with a complete destination address (target subscriber no. and target object no.).

  Note
  When using the block in the PLC of a node station, you should consider the consequences of PartnerNo = 0 If the PLC of the node station maintains both connections to higher-level subscribers as well as to lower-level stations, a message with PartnerNo = 0 is transferred to all subscribers both "up" and "down".

- **Sending process typicals**
  (Cmd01B_S, Set01W_S, Par12D_S)

  The parameter setting PartnerNo = 0 is not permitted!

  If the parameter setting is incorrect (< 1 or > 32000), an error message to this effect is entered in the diagnostic buffer (event ID B100). If the value range is correct, but the PartnerNo was not found in the administration (in DB-BasicData), an entry is also made in the diagnostic buffer (event ID B101). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

- **Receiving process typicals**
  (Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R, Dat12D_R)

  The parameter setting PartnerNo = 0 is not permitted!

  If the parameter setting is incorrect (< 1 or > 32000), an error message to this effect is entered in the diagnostic buffer (event ID B100). If the value range is correct, but the PartnerNo was not found in the administration (in DB-BasicData), an entry is also made in the diagnostic buffer (event ID B101). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

  If the PLC receives a message for the object set here, the system checks whether the source subscriber number in the message is identical to the PartnerNo set here. If they are different, the received information is discarded. An error message to this effect is entered in the diagnostic buffer (event ID B130).

PartnerObjectNo

Name:	**PartnerObjectNo**
Declaration:	INPUT
Data type	INT
Default:	0
Explanation	Object number of the partner
	Range of values: 0 or 1 ... 32000
	The number of the object (= DB number) on the partner with which the FB communicates, i.e. to which the FB sends data or from the FB receives data, must be specified.
	Point to note when PartnerObjectNo = 0 in various typical classes:

- **Sending process typicals**
  (Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Dat12D_S)

  Setting the parameter to 0 is useful, if PartnerNo = 0 was set. If the PartnerObjectNo is missing, there must be a list on the partner PLC from which the missing object number can be recognized (see FC ListGenerator).

  If the subscriber specified by PartnerNo is an ST7cc control center, the PartnerObjectNo does not need to be specified in the FB because there are no DBs as destination objects in ST7cc as there are in a CPU. ST7cc decodes its messages solely based on the source address in the message.

- **Receiving process typicals**
  (Cmd01B_R, Set01W_R, Par12D_R)

  The parameter setting PartnerObjectNo = 0 is necessary in the following situations:
  - The partner is not an S7-CPU; in other words, there is no object = DB number. This is, for example, the case when the partner is an ST7cc control center.
  - There is more than one partner (PartnerNo = 0) that wants to send data to this typical. The corresponding objects of these partners will then generally have different numbers; in other words, no unique number can be specified here.

  If the PLC receives a message for the object set here, and PartnerObjectNo is greater than 0, the system checks whether the source object number in the message is identical to the PartnerObjectNo set here. If they are different, the received information is discarded. An error message to this effect is entered in the diagnostic buffer (event ID B131).

  This check is not made if PartnerObjectNo = 0. Regardless of the sender object, each message addressed to the object is also passed on to the object.

- **Sending process typicals**
  (Cmd01B_S, Set01W_S, Par12D_S)

  The parameter setting PartnerObjectNo = 0 is not permitted!

  If the parameter setting is incorrect (< 1 or > 32000), an error message to this effect is entered in the diagnostic buffer (event ID B102). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

- **Receiving process typicals**
  (Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R, Dat12D_R)

  The parameter setting PartnerObjectNo = 0 is not permitted!

  If the parameter setting is incorrect (< 1 or > 32000), an error message to this effect is entered in the diagnostic buffer (event ID B102). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

  If the PLC receives a message for the object set here, the system checks whether the source object number in the message is identical to the PartnerObjectNo set here. If they are different, the received information is discarded. An error message to this effect is entered in the diagnostic buffer (event ID B131).

Enabled

Name:	**Enabled**	
Declaration:	INPUT	
Data type	BOOL	
Default:	TRUE	
Explanation	Enable block processing	
	Range of values:	TRUE or FALSE
		No parameter specified: Default TRUE is valid
		Input I 0.0 ... I n.7
		Bit memory M 0.0 ... M n.7 L 0.0 ... L n.7 DBm.DBX 0.0 ... n.7
		Data bit

Whether or not processing of the block is enabled must be specified.

If processing is enabled, all the functions of the FB execute.

The response is different if processing has not been enabled:

- **Processing of process typicals not enabled**
  (Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Set01W_R, Par12D_R, Dat12D_S)

  If processing is not enabled, the FB can only communicate at the organizational level; in other words, ORG messages can be sent and received. A query is, for example, answered, however the reply message contains the data valid at the time the function was disabled.

  Note
  The response described here does not apply to Cmd01B_R (see ST7 command typical FB Cmd01B_R (Page 255))!

- **Processing of operator typicals not enabled**
  (Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R, Set01W_S, Par12D_S, Dat12D_R)

  If processing is not enabled, the FB can only communicate at the organizational level; in other words, ORG messages can be sent and received. A request can, for example, still be sent and the answer received, the received information is, however, not output to the outputs.

  You will find the relevant outputs under "Enable" in the description of the individual data point typicals.

  Note
  The response described here does not apply to Cmd01B_S (see ST7 command typical FB Cmd01B_S (Page 253))!

ImageMemory

Name:	**ImageMemory**		
Declaration:	INPUT		
Data type	BOOL		
Default:	TRUE		
Explanation	Image memory principle for spontaneous data transmission		
	Range of values:	TRUE or FALSE	
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified: Default value TRUE is valid.	

You need to specify whether or not the message will be transmitted according to the image memory principle or the send buffer principle.

The image memory method reduces the memory required for storing messages on the TIM and produces as little message traffic on the WAN as possible. The default TRUE was selected because the image memory principle is the best choice in practice for most data transmissions.

In general, as the user you only need to change the default setting of the image memory parameter with a few objects, namely objects whose data changes must be stored on the TIM and sent to the partner singly, for example alarms with time stamp.

The parameter is used by the following process typicals:
Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Set01W_R, Par12D_R, Dat12D_S

Conditional

Name:	**Conditional**		
Declaration:	INPUT		
Data type	BOOL		
Default:	TRUE		
Explanation	Conditional spontaneous data transmission		
	Range of values:	TRUE or FALSE	
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified: Default value TRUE is valid.	

You will find information on the parameter assignment in the Unconditional parameter.

The parameter is used by the following process typicals:
Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Set01W_R, Par12D_R, Dat12D_S

Unconditional

Name:	**Unconditional**		
Declaration:	INPUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Unconditional spontaneous data transmission.		
	Range of values:	TRUE or FALSE	
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7

No parameter specified: Default value TRUE is valid.

Note on the use of the Conditional and Unconditional parameter settings: With the two parameters Conditional and Unconditional, you can decide whether a message is transmitted by the TIM immediately when data changes or at a later point in time.

1. If the transmission does not need to be made immediately, set the parameters as follows:
   Conditional = TRUE
   Unconditional = FALSE
2. If you require immediate transmission, the parameter combination should be:
   Conditional = FALSE
   Unconditional = TRUE

The decision for immediate or later transmission only relates to dial-up networks. On a dedicated line, the transmission is always immediate even if the combination of Conditional and Unconditional is set to "not immediately". The default of the two parameters was chosen so that a message is not transmitted immediately (combination 1). On dedicated lines, you as the user do not need to make changes to the two parameter settings. Only in a dial-up network, do you need to decide which objects are so important that an immediate transmission is necessary if there is a change in the acquired data for the object. Only then do you need to change Conditional to FALSE and Unconditional to TRUE, for example for an object with alarms.

The parameter is used by the following process typicals:
Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Set01W_R, Par12D_R, Dat12D_S

Permanent

Name:	**Permanent**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Permanent data transmission. This parameter has no significance. The functionality of permanent data transmission is not supported by the TIM.

Note

The "Permanent" parameter is no longer implemented, it has been retained to ensure compatibility.

TimeStamp

Name:	**TimeStamp**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Time stamp Range of values: TRUE or FALSE Here, you specify whether the message is transferred with the time stamp. This is only possible if the time provided by the local TIM is available on the PLC. For more detailed information, refer to the description of FC TimeTask. If no parameter is specified, the default is FALSE; in other words, data is transmitted without a time stamp. The following data point typicals use the parameter: Bin04B_S, Ana04W_S, Cnt01D_S/Cnt04D_S, Dat12D_S, Set01W_R, Par12D_R **Note** With FB-Ana04W_S, remember that the "Timestamp" parameter depends on the "MeanValueGeneration" parameter (seeST7 analog value typical FB Ana04W_S (Page 236)).

NewData

Name:	**NewData**		
Declaration:	OUTPUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Receive new data.		
	Range of values:	Output	Q 0.0 ... Q n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7

No parameter specified	Default value FALSE is valid

The NewData output is intended for user-specific further processing, for example to react in a specific way to receipt of new data.

The following data point typicals use the parameter:
Bin04B_R, Ana04W_R, Cnt01D_R/Cnt04D_R, Cmd01B_R, Set01W_S, Set01W_R, Par12D_S, Par12D_R, Dat12D_R

Whenever the FB has received new data and has output it to the outputs for the specific typical, the NewData output is set to TRUE for one OB1 cycle. You will find the relevant outputs for the specific typical under "NewData" in the description of the individual data point typicals.

With the data point typicals "Set01W_R" and "Par12D_R", NewData is also set to TRUE for one OB1 cycle if there is a new local entry in the Local = 1 state.

If you do not require the parameter, simply leave it open.

2.5.2 ST7 binary information typical FB Bin04B_S

Function

Send 4 bytes of messages/binary information

Explanation of the parameters

Names: **PartnerNo**
PartnerObjectNo
Enabled
ImageMemory
Conditional
Unconditional
Permanent
TimeStamp

Refer to Parameters you require often (Page 223).

Name: **InputByte_1 … _4**
Declaration: INPUT
Data type BYTE
Default: 0 (B#16#0)
Explanation Input byte

Range of values:	Input byte	IB0 ... IBn PIB0 ... PIBn
	Memory bytes	MB0 ... MBn LB0 ... LBn
	Data bytes	DBm.DBB0 ... n
	No parameter specified: Default value 0 is valid.	

You can specify the bytes from where the binary information such as status messages, alarms etc. is taken by the FB to be transferred in the data messages. Input bytes from the process input image, I/O bytes directly from digital input modules, data bytes from a data block and memory bytes can be mixed as required.

If you do not require parameters, simply leave them open. The value 0 is transferred for these bytes in the message.

Name:	**DisableMask**
Declaration:	INPUT
Data type	DWORD
Default:	0 (2#0)
Explanation	Disable mask
Range of values:	as 32-bit binary number 2#0 ... 2#11111111_11111111_11111111_11111111 as 32-bit hex number DW#16#0 ... DW#16#FFFF_FFFF no parameter specified: Default value 0 (2#0) is valid

A 1 must be entered in the bit pattern that the position of each input to be disabled; 0 is entered for the other inputs. The assignment of the 32 inputs of the parameters InputByte_1 through InputByte_4 to the 32 bits in the bit pattern of the DisableMask parameter is shown in the following table.

A disabled input always has the value 0 in the message.

	InputByte_1								**InputByte_2**								**InputByte_3**								**InputByte_4**							
Bit	.7	.6	.5	.4	.3	.2	.1	.0	.7	.6	.5	.4	.3	.2	.1	.0	.7	.6	.5	.4	.3	.2	.1	.0	.7	.6	.5	.4	.3	.2	.1	.0
2#	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
DW#16#	_				_				_				_				_				_				_				_			

Name:	**InversionMask**
Declaration:	INPUT
Data type	DWORD
Default:	0 (2#0)
Explanation	Inversion mask

Range of values: as 32-bit binary number
2#0 ... 2#11111111_11111111_11111111_11111111
As 32-bit binary number
DW#16#0 ... DW#16#FFFF_FFFF
No parameter specified: Default value 0 (2#0) is valid.

A 1 must be entered in the bit pattern that the position of each input to be inverted; 0 is entered for the other inputs. The assignment of the 32 inputs of the parameters InputByte_1 through InputByte_4 to the 32 bits in the bit pattern of the InversionMask parameter is shown in the following table.

The inversion of input signals can, for example, be useful when using a mixture of sensors operating on the open and closed circuit principle.

	InputByte_1								InputByte_2								InputByte_3								InputByte_4							
Bit	.7	.6	.5	.4	.3	.2	.1	.0	.7	.6	.5	.4	.3	.2	.1	.0	.7	.6	.5	.4	.3	.2	.1	.0	.7	.6	.5	.4	.3	.2	.1	.0
2#	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
DW#16#	_				_				_				_				_				_				_				_			

2.5.3 ST7 binary information typical FB Bin04B_R

Function

Receive 4 bytes of status/binary information

Explanation of the parameters

Names: **PartmerNo**
PartmerObjectNo
Enabled

Refer to Parameters you require often (Page 223).

Name: **OutputByte_1 … _4**
Declaration: OUTPUT
Data type BYTE
Default: 0 (B#16#0)
Explanation Output byte

Range of values:	Output byte	QB0 ... QBn PQB0 ... PQBn
	Memory bytes	MB0 ... MBn LB0 ... LBn
	Data bytes	DBm.DBB0 ... n

Where the binary information such as status messages, alarms etc is to be output can be selected byte by byte. Output bytes in the process output image, I/O bytes directly on digital output modules, data bytes of a data block and memory bytes can be mixed as required.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

If you do not require parameters, simply leave them open.

Name: **NewData**

Refer to Parameters you require often (Page 223).

Explanation of the typical-specific outputs: Whenever the FB has received new data and has output it to the output bytes OutputByte_1 through OutputByte_4, the NewData output is set to TRUE for one OB1 cycle.

2.5.4 ST7 analog value typical FB Ana04W_S

Function

Send 4 analog values (16-bit value in the INT format).

FB Ana04W_S transfers the 4 analog values:

- As instantaneous values
  At the time of the transmission, the currently pending analog value is acquired and transferred to the partner.
  or
- As mean values
  The pending analog value is accumulated at selectable intervals. At the time of the transmission, a mean value is formed from the total value and transferred to the partner.

Note

The processing parameters such as threshold, smoothing factor etc. exist only once in a typical. These parameters apply to all 4 analog values in common; in other words, it is not possible to set the parameters for the individual analog values. For this reason, each typical should only acquire analog values that can be processed in the same way.

Explanation of the parameters

Names: **PartnerNo**
PartnerObjectNo
Enabled
ImageMemory
Conditional
Unconditional
Permanent

Refer to Parameters you require often (Page 223).

Name: **TimeStamp**

Declaration, data type , default and range of values: Refer to Parameters you require often (Page 223).

Explanation Time stamp

Here, you specify whether the message is transferred with the time stamp. The prerequisite is that the time provided by the local TIM is available on the PLC. For more detailed information, refer to the description of FC TimeTask.

The following applies to the time stamp in the message;

- MeanValueGeneration = FALSE (instantaneous values are transmitted in the message)
  The time stamp in the message is identical to the time of acquisition of the instantaneous values contained in the message.
- MeanValueGeneration = TRUE (the message contains mean values)
  The time stamp is identical to the time at which the mean value calculation period was completed. The start of the mean value calculation period is not included in the message. This is, however, identical to the time stamp of the previously transferred mean value message.

If no parameter is specified, the default is FALSE; in other words, data is transmitted without a time stamp.

Name: **ThresholdIntegration**
Declaration: INPUT
Data type BOOL
Default: FALSE
Explanation Threshold value processing according to the integration principle.

Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid.

With this parameter, you can specify whether the integration principle is used in threshold value processing.

If no parameter is specified, the default is FALSE; in other words, threshold values are processed without integration. This corresponds to the previous ST1 procedure. In this case, you can also expect less traffic on the telecontrol line and locally between CPU and TIM (over the MPI bus or party line).

Note

When MeanValueGeneration = TRUE, i.e. the analog values are sent as mean values, the ThresholdIntegration parameter has no meaning.

Name:	**ZeroLimitation**
Declaration:	INPUT
Data type	BOOL
Default:	TRUE
Explanation	Zero limitation.

Range of values:	TRUE or FALSE No parameter specified: Default value TRUE is valid.

This parameter allows you to specify whether negative values should be suppressed and replaced with the value 0.

If no parameter is specified, the default is TRUE. This means that the lowest value is limited to 0.

Name:	**TriggerInput**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Trigger input.

Range of values:	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
	No parameter specified: Default value FALSE is valid.	

If required, this parameter can be used to specify an input over which the user can trigger the transmission *) of the analog value message at any time (signal edge from 0 to 1).

Example:
Time-driven analog value transmission with time stamp for supplying an analog value archive in the control center. Note: To prevent these messages with time stamps from being overwritten when saving on the station TIM, the ImageMemory parameter must be set to FALSE.

If the block calculates mean values, the duration of the calculation period is defined by the TriggerInput input. The current period is ended and a new period begun each time a transmission is triggered by this input. The interval between two message triggers therefore determines the duration of the mean value calculation period.

FC Trigger can be used for time-driven triggering of a transmission over TriggerInput (for more detailed information, refer to the description of this block).

If you do not require the parameter, simply leave it open. Message transmission should then be triggered based on the ThresholdValue and ThresholdIntegration threshold parameters.

*) TriggerInput actually only triggers transmission indirectly. With a 0/1 edge at TriggerInput, the message is put together with its current values/mean values and transferred to the local TIM. The TIM is responsible for the actual transmission to the partner. Transmission is immediate over a dedicated line/wireless link; with a dial-up connection, it is possible that the message is saved first on the TIM and sent at a later point in time (for example, because the message is marked as a "conditional spontaneous" message; see the Conditional parameter).

Name:	**MeanValueGeneration**	
Declaration:	INPUT	
Data type	BOOL	
Default:	FALSE	
Explanation	Mean value generation.	
	Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid.

With this parameter, you can specify whether the analog values to be acquired are transferred as mean values.

If you select mean value generation, the currently pending analog value is acquired cyclically and accumulated. The acquisition cycle depends on the SamplingPeriod parameter (for example 500 ms, see also the description of this parameter). The mean value is calculated from the accumulated values as soon as a transmission is triggered over the TriggerInput input. Following this, the accumulation starts again so that the next mean value can be calculated.

The mean value can also be calculated if the transmission of the analog value message is triggered by a general or single request. The duration of the mean value calculation period is then the time from the last transmission (for example triggered over TriggerInput) to the time of the general or single request. Once again, the accumulation restarts so that the next mean value can be calculated.

If the acquired analog value is above or below the permitted range (7FFFH bzw. 8000H), this value can either be taken into account immediately in the calculation of the mean value or it can be suppressed for a specific period for the calculation of the mean value. The required response can be decided with the FaultSuppressionTime parameter:

FaultSuppressionTime = 0
Acquisition of a value above or below the allowed range results in an immediate cancelation of the mean calculation. The value 7FFFH or 8000H is saved as an invalid mean value for the current mean value calculation period and sent when the next analog value message is triggered. The calculation of a new mean value is then started. If the analog value remains above or below the permitted range, this new value is again saved immediately as an invalid value and sent when the next message is triggered.

FaultSuppressionTime > 0
If the acquired analog value is above or below the permitted range, the bad values are excluded from the calculation of the mean value for a maximum duration as defined by the FaultSuppressionTime. If this period is exceeded, the value 7FFFH or 8000H is saved as an invalid mean value and sent when the next analog value message is triggered. This procedure is repeated in the new mean value calculation period; in other words, bad values are suppressed once again for the duration of the FaultSuppressionTime.
The FaultSuppressionTime period allows you to indirectly determine the percentage of invalid values for each mean value calculation period. For example, if the mean is calculated every 15 minutes and FaultSuppressionTime is set to 5 minutes, the mean value is only sent as invalid when more than 1/3 or 33% of the analog values acquired are above or below the permitted range in the current mean value calculation period.

If no parameter is specified, the default is FALSE; in other words, instantaneous values are acquired and transmitted.

Name:	**AnalogInput_1 ... _4**		
Declaration:	INPUT		
Data type	WORD		
Default:	0 (W#16#0)		
Explanation	Analog input word.		
	Range of values:	I/O words	PIW0 ... PIWn
		Memory words	MW0 ... MWn LW0 ... LWn
		Data words	DBm.DBW0 ... n

For each analog value to be transmitted in the data message, you can specify from where the FB will take the analog information. I/O words from analog input modules, data words from a data block and memory words can be mixed as required.

If you do not require parameters, simply leave them open. The value 0 is transferred for these analog inputs in the message.

Name:	**SamplingPeriod**
Declaration:	INPUT
Data type	INT
Default:	500
Explanation	Acquisition interval for analog inputs in ms.

Range of values:	0 ... 32767 [ms] No parameter specified: Default value 500 ms is valid.

The acquisition interval is required for the following parameters:

- For the processing of the threshold value according to the integration principle (threshold integration)
- For smoothing the analog input value (SmoothingFactor)
- For calculating the mean values (MeanValueGeneration)

The value must be selected high enough so that it is certain that a new value was acquired over the analog input. The interval to be specified has to be at least as long as the encoding time of the analog input module being used at the selected resolution (8 ... 15 bits).

If no parameter is specified, the default of 500 ms applies. This time is high enough to be applied even at the highest resolution and for analog modules with the maximum number of inputs.

If mean values are calculated, SamplingPeriod should not be less than 500 ms. If mean values are calculated over very long periods, the time must be increased as follows:

- Mean value calculation period 12 h: SamplingPeriod = 1000 [ms]
- Mean value calculation period 24 h: SamplingPeriod = 2000 [ms]

Specifying a SamplingPeriod that is too short may lead to an overflow of internal accumulation counter (must not exceed max. value 2,147,483,647 of a double integer). When an overflow is detected, the invalid mean value of 8000H is transmitted for the current mean value calculation period.

Name:	**ThresholdValue**
Declaration:	INPUT
Data type	INT
Default:	270
Explanation	Threshold value.

Range of values:	0 or 1 ... 32767 No parameter specified: Default 270 is valid (corresponds to 1%).

The encoding range of the analog value must be taken into consideration when setting the threshold value. Raw values from S7 analog inputs are always encoded in the range from 0 ... 27648 (= 0 ... 100 %) or + 27648 (= + 100%). Depending on the resolution of the analog input, the value jumps by 128 (at 8-bit resolution) or 1 (at 15-bit resolution). If the acquired analog values have a different encoding range, the threshold value should be entered according to this.

If no parameter is entered, the default value of 270 applies. This corresponds to approximately 1% of the normal S7 analog raw value range.

Point to note with ThresholdValue = 0

Changes are not checked based on the threshold value. The analog value message will only be sent in the following situations:

1. When there is a trigger over the TriggerInput input, typically a time-driven or event-driven message trigger.
2. When there is a general request to the station or a single request for the message.
3. When the analog value moves into the overflow or underflow range (7FFFH or 8000H) (possibly after the suppression time set by FaultSuppressionTime has elapsed).

Note

When MeanValueGeneration = TRUE, i.e. the analog values are sent as mean values, the ThresholdValue parameter has no meaning.

Name:	**SmoothingFactor**
Declaration:	INPUT
Data type	INT
Default:	1
Explanation	Smoothing factor.

Range of values:	1 (no smoothing) 4 (weak smoothing) 32 (medium smoothing) 64 (strong smoothing) No parameter specified: Default 1 (no smoothing) is valid.

Using the smoothing factor, quickly fluctuating analog values can be smoothed to a greater or lesser extent (depending on the factor). This may allow a narrower threshold band to be set (see ThresholdValue).

The smoothing factors are identical to the smoothing factors that are configured for some S7 analog input modules. The smoothing is handled in typical using the same formula as for input modules, described by the following:

$$y_n = \frac{x_n + (k-1)y_{n-1}}{k}$$

y_n = smoothed value in the current cycle n

y_n = acquired value in the current cycle n

k = smoothing factor

Note

When MeanValueGeneration = TRUE, i.e. the analog values are sent as mean values, the SmoothingFactor parameter has no meaning.

Name:	**FaultSuppressionTime**
Declaration:	INPUT
Data type	INT
Default:	0
Explanation	Fault suppression time in seconds.

Range of values: 0 ... 32767 [s]
No parameter specified: Default value 0 s is valid.

Transmission of an analog value located in the overflow or underflow range (7FFFH or 8000H) is suppressed for the time period specified here. The value 7FFFH or 8000H is only sent after this time has elapsed, if it is still pending. If the value returns to below 7FFFH or above 8000H again before this time elapses, it is immediately sent again as normal. The suppression time is started again for the full duration the next time 7FFFH or 8000H is acquired.

This is typically used for temporary suppression of current values that may occur when powerful pumps and motors are started. The analog input may exceed several times the maximum range under some circumstances. Suppression prevents these values from being signaled as faults in the control center system.

The suppression is adjusted to analog values that are acquired by the S7 analog input modules as raw values. These modules return the specified values for the overflow or underflow range for all input ranges (also for life-zero inputs). When the user provides specific values, fault suppression is only possible if these also adopt the values 7FFFH or 8000H when the permitted ranges are exceeded. If this is not the case, the parameter does not need to have a value entered.

The parameter can also be used in combination with the mean value calculation temporary suppression of the values 7FFFH or 8000H (see description of the MeanValueGeneration) parameter.

When no parameter is specified, the default value of 0 seconds applies. An acquired value of 7FFFH or 8000H is then sent immediately when it is first detected or, with mean value calculation, as an invalid mean value for the current mean value calculation period.

2.5.5 ST7 analog value typical FB Ana04W_R

Function

Receive 4 analog values (16-bit value in the INT format).

Explanation of the parameters

Names: **PartnerNo**
PartnerObjectNo
Enabled

Refer to Parameters you require often (Page 223).

Name:	**AnalogOutput_1 … _4**		
Declaration:	OUTPUT		
Data type	WORD		
Default:	TRUE		
Explanation	0 (W#16#0)		
	Range of values:	I/O words	PQW0 ... PQWn
		Memory words	MW0 ... MWn LW0 ... LWn
		Data words	DBm.DBW0 ... n

You can select where the individual analog values received by the FB are output. I/O words from analog output modules, data words from a data block and memory words can be mixed as required.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

If you do not require parameters, simply leave them open.

Name: **NewData**

Refer to Parameters you require often (Page 223).

Explanation of the typical-specific outputs: Whenever the FB has received new data and has output it to the outputs AnalogOutput_1 through AnalogOutput_4, the NewData output is set to TRUE for one OB1 cycle.

2.5.6 ST7 counted value typicals FB Cnt01D_S and FB Cnt04D_S

Function

[Cnt01D_S] send 1 counted value (32-bit ST1 format).

[Cnt04D_S] send 4 counted values (32-bit ST1 format).

Explanation of the parameters

Names: **PartnerNo**
PartnerObjectNo
Enabled
ImageMemory
Conditional
Unconditional
Permanent

TimeStamp

Refer to Parameters you require often (Page 223).

Name:	**GeneralTriggerCommand**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	General restore command:
Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid.

Here, you specify whether or not a counted value transmission should be triggered by a general restore command (the general restore command belongs to the organizational SINAUT system commands).

If an explicit destination subscriber number (PartnerNo > 0) is assigned in the typical, the general restore command is evaluated in the corresponding subscriber object in the central administration. If the destination subscriber number is missing (PartnerNo = 0, send 'to all'), the central system memory bit 'General restore command' is taken into account.

When the general restore command is detected, the currently accumulated counted value is transmitted regardless of other criteria that affect message transmission. The restore identifier US is inverted in this counted value.

Parameters GeneralTriggerCommand and TriggerInput can be used together. Transmission is then triggered by a signal edge change from 0 to 1 at the TriggerInput as well as when a general restore command is received.

If no parameter is specified, the default is FALSE; in other words, there is no restore using the general restore command.

Name:	**TriggerInput**	
Declaration:	INPUT	
Data type	BOOL	
Default:	FALSE	
Explanation	Trigger input	
Range of values:	TRUE or FALSE	
	No parameter specified: Default value FALSE is valid	
	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

If required, this parameter can be used to specify an input over which the user can trigger (signal edge change from 0 to 1) a transmission at any time regardless of other criteria that affect message transmission. The currently accumulated counted value is transmitted. The restore identifier US is inverted in this counted value.

Note

FC Trigger is an easy way to trigger time-driven transmission of a counted value message. For more detailed information, refer to the description of the FC.

Parameters TriggerInput and GeneralTriggerCommand can be used together. Transmission is then triggered by a signal edge change from 0 to 1 at the TriggerInput as well as when an organizational restore command is received.

If no parameter is specified, the default is FALSE; in other words, there is no restore and transmission triggered over the TriggerInput input.

Name:	**[Cnt01D_S] Counter_1** **[Cnt04D_S] Counter_1 ... _4**
Declaration:	INPUT
Data type	COUNTER
Default:	-
Explanation	Number of the SIMATIC counter.
	Range of values: C0 as dummy parameter or C1 ... Cn (n depending on CPU type)
	Here, you specify the SIMATIC counter in which the pulses were counted time-driven. This takes place in the background using FC PulseCounter that is called in a cyclic interrupt OB, for example OB35. Refer to the description of FC PulseCounter and 'Time-driven SINAUT program in a cyclic interrupt OB'.
	The COUNTER data type cannot be assigned a default value. If you have used the C0 dummy parameter in the typical, the corresponding counted value is not processed.

Name:	**DifferenceValue**
Declaration:	INPUT
Data type	INT
Default:	0
Explanation	Difference value.
	Range of values: 0 or 1 ... 31768 No parameter specified: Default value 0 is valid.

When a value between 1 and 31768 is specified, the counted value is sent as soon as the difference between the current and most recently transmitted counted value reaches or exceeds the configured value.

If no parameter is specified, default value 0 applies; in other words, a counted value is only sent when a signal edge change from 0 to 1 is detected at the TriggerInput input , or when (if GeneralTriggerCommand = TRUE) an organizational. restore command is received.

Note

The difference value must be selected by the user depending on the maximum pulse rate per second. The value should not be too low otherwise there is a constant transfer of the message to the TIM. On one hand, this would cause heavy load on the MPI bus/party line but also stretch the send queue on the CPU.

Note

[Cnt04D_S]

The DifferenceValue processing parameter exists only once in the typical. This parameter applies to all 4 counted values in common; in other words, it is not possible to set the parameter for the individual counted values. When using this parameter, each typical should therefore only acquire counted values that can be processed identically.

2.5.7 ST7 counted value typicals FB Cnt01D_R and FB Cnt04D_R

Function

[Cnt01D_R] receive 1 counted value (32-bit ST1 format).

[Cnt04D_R] receive 4 counted values (32-bit ST1 format).

Explanation of the parameters

Names:	**PartnerNo**
	PartnerObjectNo
	Enabled

Refer to Parameters you require often (Page 223).

Name:	**BCD_Format**
Declaration:	INPUT
Data type	BOOL
Default:	TRUE
Explanation	Counted value output in BCD format.

Range of values:	TRUE or FALSE No parameter specified: Default value TRUE is valid.

Here, you specify the format in which counted value received at the output / outputs CountedValueOutput_... is output.

If the parameter is left open or if you specify TRUE, the counted value is output with a maximum of seven places in BCD (= S7 format; the sign decade is always 0; in other words, +). The maximum counted value that can be represented is therefore restricted to 9,999,999.

If you do not require BCD format (BCD_Format = FALSE), the counted value is output as a 32-bit integer and this is also always positive. The maximum counted value that can be represented is then 2,147,483,647.

If the maximum counted value that can be represented is exceeded, the counted value starts again at 0 and counting continues in the positive numeric range.

Note

[Cnt04D_R]

The BCD_Format processing parameter exists only once in the typical. This parameter applies to all 4 counted values in common; in other words, it is not possible to set the parameter for the individual counted values. When using this parameter, each typical should therefore only output counted values that can have parameters set for an identical output format.

Name:	**CntValInvalid**		
Declaration:	OUTPUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Counted value invalid.		
	Range of values:	Output	Q 0.0 ... Q n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7

At the CntValInvalid output, the FB indicates whether the last received counted value was invalid (with Cnt04D_R, this counts as a group display for all four counted values). In principle, the output shows the current status of the validity bit A from the most recently received counted value in inverted form.

On the one hand, the output serves as an error indicator. On the other, the output is intended for user-specific further processing. For example, the user may wish to react to invalidity by correcting the counted value at CountedValueOutput_... by adding counter pulses that may have been lost.

If you do not require the parameter, simply leave it open.

Note

[Cnt04D_R]

Although all 4 counted values in the message have their own validity bit, only the validity bit A of the first counted value in the last received message is evaluated for the status at the CntValInvalid output. This status, however, applies to all 4 counted values, since all the counted values in the message always have the same validity status.

Note

When evaluating CntValInvalid, you should take into account the fact that it might only be possible for this bit to be set for one OB1 cycle.

Name:	**RestoreStatus**		
Declaration:	OUTPUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Current status of the restore bit US in the received counted value.		
	Range of values:	Output	Q 0.0 ... Q n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified: Default value FALSE is valid.	
	At the RestoreStatus output, the FB indicates the current status of the restore bit US from the last received counted value message.		
	The output is intended for user-specific further processing. For example, the user may only wish to access the information at CountedValueOutput_... when a change has been detected at the RestoreStatus output; in other words, when the counted value has been received due to a restore, such as a local time-driven restore.		
	If you do not require the parameter, simply leave it open.		

Note

[Cnt04D_R]

Although all 4 counted values in the message have their own restore bit, only the restore bit US of the first counted value in the last received message is evaluated for the status at the RestoreStatus output. This status, however, applies to all 4 counted values since they are always restored together.

Name:	**NewData**
Refer to Parameters you require often (Page 223).	
Explanation of the typical-specific outputs	Whenever the FB has received new data and has output it to the output / outputs CountedValueOutput_..., the NewData output is set to TRUE for one OB1 cycle.

Name:	**[Cnt01D_R] CountedValueOutput_1** **[Cnt04D_R] CountedValueOutput_1 ... _4**
Declaration:	IN_OUT
Data type	DWORD
Default:	0
Explanation	Counted value output.

Range of values:	(process image) output double words	QD0 ... QDn
	Memory double words	MD0 ... MDn
	Data double words	DBm.DBD0 ... n

The counted value output is a double word in which the counted value is stored in BCD format or as a 32-bit integer (depending on the BCD_Format parameter). In BCD format, the maximum counted value that can be represented is limited to 9,999,999, if the counted value is output as a 32-bit integer, the maximum counted value that can be represented is 2,147,483,647.

The counted value is always output as a positive number; in other words, if the maximum counted value that can be represented is exceeded, the counted value starts again at 0 and counting continues in the positive numeric range.

Since the parameter is an in-out parameter (declaration IN_OUT), the value can be reset to 0 or another value at the counted value output by the user at any time. The counted value typical always adds the newly formed difference value (difference between the new and last received counted value) to the value currently output that the counted value output.

Note

Since the parameter is an in-out parameter (declaration IN_OUT), direct I/O output of the counted value to PQD0 ...PQDn is not permitted. It is also difficult to specify local bit memory with this parameter type and this should not be used.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

2.5.8 ST7 command typical FB Cmd01B_S

Function

Send 1 byte commands (1-out-of-8 ST1 format).

Note

With FB Cmd01B_S, data can only be transmitted when FC Safe is included at the end of the cyclic SINAUT program. See also 'The cyclic OB1 program for a master station' in section Cyclic SINAUT program in OB1 (Page 206).

Explanation of the parameters

Names: **PartnerNo**
PartnerObjectNo

Refer to Parameters you require often (Page 223).

Name: **Enabled**

Declaration, data type , default and range of values: Refer to Parameters you require often (Page 223).

Explanation Whether or not processing of the block is enabled must be specified.

If processing is enabled, all the functions of the FB execute.

If processing is disabled the FB only checks to see if the disabled status has been canceled. The FB cannot communicate on the organizational level in this status because FB Cmd01B_S cannot send or receive organizational messages.

Name: **CommandInputByte_HW**
Declaration: INPUT
Data type BYTE
Default: 0 (B#16#0)
Explanation Command input byte for hardware input.

Range of values:	Input byte	IB0 ... IBn PIB0 ... PIBn
	Memory bytes	MB0 ... MBn LB0 ... LBn
	Data bytes	DBm.DBB0 ... n

This command input byte is specially designed for entering commands using hardware, i.e. over digital inputs. Input using memory or data bytes is also possible, but the user must then make sure that the command at the input byte is reset, which occurs at the hardware level when the command button is released.

When input is detected, the command is transmitted if no error is detected during the 1-out-of-8 and 1-out-of-n check, and if the central enable memory bit is set. This is automatically set by FC Safe following a selected time delay set there (see FC Safe, InputDelayTime parameter).

If a 1-out-of-8 or 1-out-of-n error is detected, the entered command is no longer processed. A new command is first read in when no hardware command has been acquired in the PLC for one OB1 cycle; in other words, not only for this block but also for all other command input blocks with hardware input. The FB enters the detected 1- out-of-8- or 1-out-of-n error in the diagnostic buffer (event ID B171 or B172). The error status is also indicated over the InputError output of FC Safe (see FC Safe, InputError parameter) and continues to be indicated as long as the error remains.

Name:	CommandInputByte_SW
Declaration:	IN_OUT
Data type	BYTE
Default:	0 (B#16#0)
Explanation	Command input byte for software input.

Range of values:	Memory bytes	MB0 ... MBn
	Data bytes	DBm.DBB0 ... n

This command input byte is specially designed for entering commands using software, i.e. by the user program or at an operator panel (OP). When input is detected, the command is reset at the input byte and transmitted if no error is detected during the 1-out-of-8 and 1-out-of-n check. The central enable memory bit is ignored here because it is only intended for command input over hardware (see CommandInputByte_HW).

If a 1-out-of-8 or 1-out-of-n error is detected, the entered command is no longer processed. A new command is first read in when no software command has been acquired on the PLC for one OB1 cycle; in other words, not only for this block but also for all other command input blocks with software input. The FB enters the detected 1- out-of-8- or 1-out-of-n error in the diagnostic buffer (event ID B171 or B172). Appropriate error bits are also set in the central data block "BasicData" where they can be queried by the software. For more detailed information, refer to the description of FC Safe.

In principle it is possible to enter a new command to CommandInputByte_SW in every OB1 cycle. However, only one command per OB1 cycle is allowed and this applies to all command input blocks with software input (1-out-of-n check). An ’empty cycle’ between two consecutive software commands is therefore not necessary.

Note

The command inputs CommandInputByte_HW and CommandInputByte_SW can also be used at the same time; in other words, if you want to enter the same command over the hardware and software. If a command entry occurs at the same time over both input bytes, this is only accepted when coincidentally exactly the same command is entered over the hardware as well as the software input (the hardware entry is then processed). In all other cases the entry is rejected and an error message is entered in the diagnostics buffer (event ID B170). The error status is also indicated by the InputError output of FC Safe and appropriate error bits are set in the central data block "BasicData" where they can be queried by the software (see FC Safe).

Note

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

2.5.9 ST7 command typical FB Cmd01B_R

Function

Receive 1 byte commands (1-out-of-8 ST1 format).

Explanation of the parameters

Names: **PartnerNo**
PartnerObjectNo

Refer to Parameters you require often (Page 223).

Name: **Enabled**

Declaration, data type , default and range of values: Refer to Parameters you require often (Page 223).

Explanation Whether or not processing of the block is enabled must be specified.

If processing is enabled, all the functions of the FB execute.

If processing is disabled the FB only checks to see if the disabled status has been canceled. Any commands that are still received are not output. The FB cannot communicate on the organizational level in this status because FB Cmd01B_R cannot send or receive organizational messages.

Note

If the Enabled input can be controlled by a switch, this local disable means that no more commands are output if they are still received. Since the block is, however, not capable of sending ORG messages, it cannot report this local disable back to the partner itself. This must be implemented by the user with a separate message, for example Bin04B_S.

Name: **MultipleOutput**
Declaration: INPUT
Data type BOOL
Default: FALSE
Explanation Simultaneous output of multiple commands permitted.

Range of values: TRUE or FALSE
No parameter specified: Default value FALSE is valid.

With this parameter, you can specify whether or not several (consecutively received) commands can be output simultaneously; in other words, you specify how the block reacts when a new command is received and the previously received command is still being output (command output time has not yet elapsed or the user program has not yet reset this command).

FALSE (default):
Multiple command output is not permitted. The newly received command overwrites the output byte. Any command is still pending is therefore reset to 0 unless the new command is identical to the old one.

TRUE:
Multiple command output is permitted. A newly received command is ORed into the current output byte. The command output time is retriggered. This applies to all pending commands.

Name:	**CommandOutputTime**	
Declaration:	INPUT	
Data type	INT	
Default:	500	
Explanation	500	
	Range of values:	Command up time for command outputs in ms. No parameter specified: Default value 500 [ms] is valid.
	The specified time applies to all command outputs. If more than one output can be set at the same time (MultipleOutput = TRUE), the output time is restarted with each newly received command. This means that pending commands are retriggered. All the command outputs are reset of the same time when the output time elapses.	
	Point to note with CommandOutputTime = 0	
	A set command output is not reset by the command typical. The user program is responsible for this.	
	If no parameter is specified, an output time of 500 ms is used as the default.	

Name:	**NewData**
Refer to Parameters you require often (Page 223).	
Explanation of the typical-specific outputs	Whenever the FB has received new data and has output it to the output byte CommandOuputByte, the NewData output is set to TRUE for one OB1 cycle.

Name:	**CommandOutputByte**		
Declaration:	IN_OUT		
Data type	BYTE		
Default:	0 (B#16#0)		
Explanation	Command output byte.		
	Range of values:	(process image) output bytes	QB0 ... QBn
		Memory bytes	
		Data bytes	MB0 ... MBn
			DBm.DBB0 ... n
	To allow the command outputs to be reset both by the command typical itself as well as by the user program (went output time = 0), the parameter was declared as an IN_OUT parameter.		

Note

Since the parameter is an IN_OUT parameter, direct I/O output of the command byte to PQB0 ... PQBn is not permitted. It is also difficult to specify local bit memory with this parameter type and this should not be used.

2.5.10 ST7 setpoint typical FB Set01W_S

Function

Send 1 setpoint (16 bits) and receive current local setpoint.

Note

With FB Set01W_S, data can only be transmitted when FC Safe is included at the end of the cyclic SINAUT program.

Explanation of the parameters

Names: **PartnerNo**
PartnerObjectNo
Enabled

Refer to Parameters you require often (Page 223).

Name:	**EnterInput**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Enter input (for 'hardware' setpoint).

Range of values:	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
	No parameter specified: Default value FALSE is valid	

A setpoint at the SetpointInput can be applied over this input triggered by a signal edge change.

A signal change at EnterInput is only taken into account when the parameter ContinuousEnterFunct = FALSE. If this condition is fulfilled, the setpoint entered at SetpointInput is applied and transmitted by a signal change from 0 to 1 even if the newly entered setpoint is identical to the previously sent setpoint.

This method of applying setpoints is suitable for input at appropriate hardware, for example a console or control panel but can also be used for entering setpoints at an operator panel (OP). In the latter case, it must be possible to trigger the input by a separate function key on the OP.

If you do not require the parameter, simply leave it open.

Name: **ContinuousEnterFunct**
Declaration: INPUT
Data type BOOL
Default: FALSE
Explanation Apply setpoint continuously (for 'software' setpoint).

Range of values:	TRUE or FALSE No parameter specified: Default value TRUE is valid.

With this parameter, you can decide whether the setpoint at SetpointInput should be continuously read in and changes checked. The change evaluation is made by comparing the current with the last setpoint that was sent.

This method of applying a setpoint is suitable for input by appropriate software but can also be used for entering setpoints at an operator panel (OP) if it does not have a separate function key that can be used to trigger the input.

If you do not require the parameter, simply leave it open.

Name: **SetpointInput**
Declaration: INPUT
Data type WORD
Default: 0 (W#16#0)
Explanation Setpoint input word.

Range of values:	Input words	IW0 ... IWn PIW0 ... PIWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n

How a setpoint available at SetpointInput is processed depends on whether it is a hardware or software input. The user specifies the type of input with the ContinuousEnterFunct parameter:

- ContinuousEnterFunct = FALSE (= hardware input)
  The setpoint at SetpointInput is only read in as long as a 1 signal is detected at EnterInput. The setpoint that is read in is then transmitted if no error is detected during the 1-out-of-n check, and if the central enable memory bit is set. This is automatically set by FC Safe following a selected time delay set there (see FC Safe, InputDelayTime parameter). The next setpoint is first read in by the FB when a 0 signal is detected for at least one OB1 cycle at EnterInput.

  If a 1-out-of-n error is detected the next time a value is applied to the hardware input, the entered setpoint is no longer processed. A new setpoint is first read in when no hardware input has been acquired in the PLC for one OB1 cycle; in other words, not only for this block but also for all other command and setpoint input blocks with hardware input. The FB enters the detected 1-out-of-n error in the diagnostic buffer (event ID B172). The error status is also indicated over the InputError output of FC Safe (see FC Safe, InputError parameter) and continues to be indicated as long as the error remains.

- ContinuousEnterFunct = TRUE (= software input)
  The setpoint at SetpointInput is read in continuously and checked for changes. The change evaluation is made by comparing the current with the last setpoint that was sent. The setpoint is sent immediately every time a change occurs unless the 1-out-of-n check detects an error. Without the setpoint having changed, a new transmission of the software setpoint can be triggered over the SendSoftSetpoint input (see below).

  While for hardware input an empty cycle must be detected before a new setpoint can be sent by the block, for software input a new setpoint can be transmitted in every OB1 cycle. This is possible only when there is no other software setpoint or software command in this cycle. Otherwise a 1-out-of- n error is detected.
  If a 1-out-of-n error is detected during the software input, the entered setpoint is no longer processed. A new setpoint is first read in when no software input has been acquired in the PLC for one OB1 cycle; in other words, not only for this block but also for all other command and setpoint blocks with software input. The FB enters the detected 1-out-of-n error in the diagnostic buffer (event ID B172). Appropriate error bits are also set in the central data block "BasicData" where they can be queried by the software. For more detailed information, refer to the description of FC Safe.

Name: **ReturnedSetpoint**
Declaration: OUTPUT

Data type	WORD		
Default:	0 (W#16#0)		
Explanation	Output word for a returned setpoint.		
	Range of values:	Output words	QW0 ... QWn PQW0 ... PQWn
		Memory words	MW0 ... MWn LW0 ... LWn
		Data words	DBm.DBW0 ... n

The partner object receiving the setpoint reports back the currently valid local setpoint. This value is displayed at the ReturnedSetpoint output. If the partner object is set to 'local' and if an input is made there, then the setpoint changed locally is displayed here at ReturnedSetpoint.

After startup of the local or partner PLC, or after restoring a connection, an automatic general request ensures that the current, local, valid setpoint is displayed at ReturnedSetpoint.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

If you do not require the parameter, simply leave it open.

Name:	**LocalOperation**		
Declaration:	OUTPUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Return message from the partner object: Object is set to local operation.		
	Range of values:	Output	Q 0.0 ... Q n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified: Default value FALSE is valid.	

A setpoint can also be set locally at the partner object that receives the setpoint. The partner object then must be set to 'local' at the Local input parameter (see FB Set01W_R below). The current status of the Local input parameter is reported by the partner object and displayed here at the LocalOperation output.

After startup of the local or partner PLC, or after restoring a connection, an automatic general request ensures that the current, local, valid status is displayed at LocalOperation.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

If you do not require the parameter, simply leave it open.

Name:	**NewData**
Refer to Parameters you require often (Page 223).	
Explanation of the typical-specific outputs	Whenever the FB has received new data and has output it to the outputs ReturnedSetpoint or LocalOperation, the NewData output is set to TRUE for one OB1 cycle.

Name:	**SendSoftSetpoint**		
Declaration:	IN_OUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Trigger input for resending the last (software) setpoint.		
	Range of values:	Bit memory	M 0.0 ... M n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified: Default value FALSE is valid.	
	See also SetpointInput parameter.		
	If you do not require the parameter, simply leave it open.		

Note

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

2.5.11 ST7 setpoint typical FB Set01W_R

Function

Receive or enter 1 setpoint locally (16 bits) and send the current, locally valid setpoint.

Explanation of the parameters

Names:	**PartnerNo**
	PartnerObjectNo
	Enabled
	ImageMemory
	Conditional
	Unconditional
	Permanent

TimeStamp

Refer to Parameters you require often (Page 223).

Name:	**Local**	
Declaration:	INPUT	
Data type	BOOL	
Default:	FALSE	
Explanation	Enable block processing.	
	Range of values:	TRUE or FALSE
		No parameter specified: Default value FALSE is valid
		Input I 0.0 ... I n.7
		Bit memory M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit DBm.DBX 0.0 ... n.7

This input is used to enable local input of a setpoint over LocalSetpointInput. A setpoint sent for example by the master is not accepted by the object as long as Local = TRUE.

The current status of the Local input is transmitted to the partner together with a copy of the setpoint which is currently being output at SetpointOutput (setpoint mirroring).

Bumpless switchover:

- When there is a switchover from Local = 0 to Local = 1, the last values at ParameterOutput are held until new parameter values are entered over LocalParameterInput.
- When there is a switchback from Local = 1 to Local = 0, the last values at ParameterOutput are held until the block receives new parameter values from the remote partner.

Note

Please read the note on the ContinuousEnterFunct parameter.

Name:	**EnterInput**	
Declaration:	INPUT	
Data type	BOOL	
Default:	FALSE	
Explanation	Enter input for local setpoint input.	
	Range of values:	TRUE or FALSE
		No parameter specified: Default value FALSE is valid

Input	I 0.0 ... I n.7
Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
Data bit	DBm.DBX 0.0 ... n.7

A setpoint at the LocalSetpointInput can be applied over this input triggered by a signal edge change.

A signal change at EnterInput is only taken into account when the value TRUE is set at the Local input parameter and ContinuousEnterFunct = FALSE. If these conditions are fulfilled, a signal change from 0 to 1 causes the setpoint at LocalSetpointInput to be applied and output at SetpointOutput.

This method of applying setpoints is suitable for input at appropriate hardware, for example a console or control panel but can also be used for entering setpoints at an operator panel (OP). In the latter case, it must be possible to trigger the input by a separate function key on the OP.

If you do not require the parameter, simply leave it open.

Name: **ContinuousEnterFunct**
Declaration: INPUT
Data type BOOL
Default: FALSE
Explanation Continuous local setpoint acquisition.

Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid.

With this parameter, you can decide whether the setpoint at LocalSetpointInput should be continuously read in and changes checked. The change evaluation is made by comparing the current with the last mirrored setpoint. The value is only read when the Local input parameter is set to TRUE.
If an array is detected, this is output immediately at SetpointOutput.

This method of acquiring a setpoint is suitable for input by appropriate software but can also be used for entering setpoints at an operator panel (OP) if it does not have a separate function key that can be used to trigger the input.

If you do not require the parameter, simply leave it open.

Note

When ContinuousEnterFunct = TRUE, the value available at LocalSetpointInput is entered immediately and passed to SetpointOutput when the 1 signal at the Local input is detected if the local input value differs from the last returned setpoint at this point in time!

Name: **LocalSetpointInput**
Declaration: INPUT

Data type	WORD		
Default:	0 (W#16#0)		
Explanation	Local setpoint input word.		
	Range of values:	Input words	IW0 ... IWn PIW0 ... PIWn
		Memory words	MW0 ... MWn LW0 ... LWn
		Data words	DBm.DBW0 ... n
		No parameter specified: Default value 0 is valid.	

A value at LocalSetpointInput is only adopted if the Local input parameter is set to TRUE. If this condition is met, how a pending setpoint is processed depends on whether it is a hardware or software input. The user specifies the type of input with the ContinuousEnterFunct parameter:

- ContinuousEnterFunct = FALSE (= hardware input)
  The setpoint at LocalSetpointInput is only read in when a signal change from 0 to 1 is detected at EnterInput. The setpoint entered locally is output over the output set with SetpointOutput and transferred to the partner for display. A further setpoint is then only read in by the FB when a 0 signal was detected at EnterInput for at least one OB1 cycle.
- ContinuousEnterFunct = TRUE (= software input)
  The setpoint at LocalSetpointInput is read in continuously and checked for changes. The change evaluation is implemented by comparing the current value with the last valid setpoint; in other words, the value stored as the returned setpoint. Every time there is a change, the setpoint is passed immediately to the output specified by SetpointOutput and sent to the partner for display.
  While for hardware input an empty cycle must be detected before a new setpoint can be read by the block, for software input a new setpoint can be entered in every OB1 cycle.

If you do not require the parameter, simply leave it open.

Name:	**SetpointOutput**		
Declaration:	OUTPUT		
Data type	WORD		
Default:	0 (W#16#0)		
Explanation	Setpoint output word.		
	Range of values:	Output words	QW0 ... QWn PQW0 ... PQWn
		Memory words	MW0 ... MWn LW0 ... LWn
		Data words	DBm.DBW0 ... n

The setpoint sent by the partner object or the setpoint entered locally at LocalSetpointInput is output to the output word specified here in SetpointOutput.

Name:	**NewData**
Refer to Parameters you require often (Page 223).	
Explanation of the typical-specific outputs	Whenever the FB has received a new setpoint from the partner object and has output it to SetpointOutput, the NewData output is set to TRUE for one OB1 cycle. This also applies when there is new local input when Local = 1.

2.5.12 ST7 parameter typical FB Par12D_S

Function

Send 1 to 12 parameter values (each 1 double word) and receive back the current, locally valid parameter values.

Note

With FB Par12D_S, data can only be transmitted when FC Safe is included at the end of the cyclic SINAUT program.

The content of each double word may be a value in double word format (e.g. DINT, REAL etc.); it can also be a mixture of other formats which together form a double word, for example,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

The data area to be sent is defined for the ParameterInput parameter in the form of an Any pointer. This data area must be within a data block and its length can vary between 1 and 12 data double words. The data area sent to the partner or the parameter values entered locally at the partner are returned from there and output here at ReturnedParameter. This output area (defined by an Any pointer) must also be within a data block and its length must match that defined for ParameterInput.

Separate data areas are normally specified for ParameterInput and ReturnedParameter. This makes it easy to recognize the most recently entered values and the current, locally valid values. However, it is also possible to specify the same data area for both parameters. The two areas then overlap 100% and therefore always match. In this case, you can no longer distinguish the difference between what has been entered most recently and what is locally valid. When returned values are not needed, there is no need to specify a data area for ReturnedParameter.

Even when separate areas are specified for ParameterInput and ReturnedParameter, it is still possible to ensure that the ParameterInput area and the ReturnedParameter value always match. This be done manually from case to case with the ApplyRemoteParamMan input or automatically by setting the ApplyRemoteParamAuto parameter to TRUE.

A parameter can also be set locally at the partner object that receives the parameter. The partner object then must be set to ’local’ at the Local input parameter (see FB Par12D_R below). The current status of the Local input parameter is reported by the partner object and displayed here at the LocalOperation output. As long as the partner object is set to 'local', no parameters are accepted there from other locations.

Transmission of the data area defined by ParameterInput can be triggered in four ways:

- With the input parameter EnterInput
  You should use this input parameter when the data area defined at ParameterInput is entered over hardware (digital and analog input modules). EnterInput must then be connected to a button on a console or panel over a digital input. The transmission of the entered values is then triggered by pressing this button. The entire data area specified by ParameterInput is always transmitted.
- With the input parameter ContinuousEnterFunct = TRUE
  You can use this parameter setting when the parameter is entered by software, for example at an operator panel (OP). There is a constant check for changes. When a change is detected in the data area defined with ParameterInput, the data double words that have changed since the last transmission are transmitted (see note).
- With the input parameter Release
  You can use this input parameter when the parameter is entered by software, for example on an OP. The Release input should then be operated by a function key on the OP. Changes are checked when a 1 signal is detected at the Release input. The data double words from the data area defined with ParameterInput that have changed since the last transmission (see note) are transmitted.
- With the input parameter RetransmitAll
  You can use this input parameter when the parameter is entered by software, for example on an OP. The RetransmitAll input should then be operated by a function key on the OP. When a 1 signal is detected at the RetransmitAll input, the entire data area defined by ParameterInput is transmitted. Changes are not checked.

Note

When the changed data area only is transmitted, this area consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

Example:
The area to be read is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word inclusive.

Note

When only changed data is transmitted and the data area contains values in double word format, the user is responsible for ensuring that these double word values are actually located in one of the maximum 12 double words of the data area to be acquired. Distribution over two consecutive data double words could otherwise lead to the transmission of only one word of the double word value (high or low word) because a change has occurred in only that particular word. The missing word could lead to processing problems on the partner that receives this value.

Explanation of the parameters

Names: **PartnerNo**
PartnerObjectNo
Enabled

Refer to Parameters you require often (Page 223).

Name: **EnterInput**
Declaration: INPUT
Data type BOOL
Default: FALSE
Explanation Enter input.

Range of values:		
	TRUE or FALSE	
	No parameter specified: Default value FALSE is valid	
	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

The transmission of the parameter value at ParameterInput can be triggered over this input by a signal edge change.

A signal change at EnterInput is only taken into account when the parameter ContinuousEnterFunct = FALSE. If this condition is fulfilled, a transition from 0 to 1 causes the parameter values specified at ParameterInput to be entered and transmitted. Changes are not checked. The entire data area specified by ParameterInput is always transmitted.

This method of transmission triggering is suitable for input with appropriate hardware, for example at a console or control panel. For more detailed information and related parameters, refer to the section Function.

If you do not require the parameter, simply leave it open.

Data checks:

- The parameters that are read in are then transmitted if no error is detected during the 1-out-of-n check, and if the central enable memory bit is set. This is automatically set by FC Safe following a selected time delay set there (see FC Safe, InputDelayTime parameter). The input area is then only read in by the FB when a 0 signal was detected at EnterInput for at least one OB1 cycle.
- When a 1-out-of-n error is detected at the "hardware" input, the entered parameters are no longer processed. New parameters are read in again only when no "hardware" input has been acquired in the PLC for one OB1 cycle; in other words, not only for this block but also for all other command, setpoint and parameter blocks with "hardware" input. The FB enters the detected 1-out-of-n error in the diagnostic buffer (event ID B172). The error status is also indicated over the InputError output of FC Safe (see FC Safe, InputError parameter) and continues to be indicated as long as the error remains.

Name:	**ContinuousEnterFunct**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Continuous change checking.

Range of values: TRUE or FALSE
No parameter specified: Default value FALSE is valid.

With this parameter, you can decide whether the parameter values at ParameterInput should be continuously read in and changes checked. The change evaluation is made by comparing the current with the last values that were sent. Only changed values are sent. If more than one change is detected, the block sends the data area in which all changed parameter values are located.

A new transmission of the parameter values can be triggered over the RetransmitAll input (see below) even when the parameter entries have not changed.

This method of transmission triggering is suitable when the parameter values are entered in the ParameterInput area by software, but can also be used for entering the parameters from an operator panel (OP) when the OP has no separate function key with which to trigger transmission. For more detailed information and related parameters, refer to the section Function.

If you do not require the parameter, simply leave it open. The default value FALSE then applies; in other words, the parameter values at ParameterInput are not read continuously and evaluated for changes.

Data checks:

- The parameters read in are only transmitted if no error is detected during the 1-out-of-n check. While for "hardware" input (see EnterInput) an empty cycle must be detected before new parameter values can be sent from the block, for "software" input new parameter values can be transmitted in every OB1 cycle. This assumes that there is no other "software" entry at another block in this cycle. Otherwise a 1-out-of-n error is detected.
- When a 1-out-of-n error is detected at the "software" input, the entered parameters are no longer processed. New parameters are read in again only when no "software" input has been acquired in the PLC for one OB1 cycle; in other words, not only for this block but also for all other command, setpoint and parameter blocks with "software" input. The FB enters the detected 1-out-of-n error in the diagnostic buffer (event ID B172).

Note

The changed data area that is transmitted consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

Example:
The area to be read is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word inclusive.

Name:	**ApplyRemoteParamAuto**
Declaration:	IN
Data type	BOOL
Default:	FALSE
Explanation	Automatic synchronization of the input area with the returned area. Range of values: TRUE or FALSE No parameter specified: Default value FALSE is valid. If ApplyRemoteParamAuto = TRUE, the input area ParameterInput is automatically synchronized with the ReturnedParameter area. All the parameter values from the ReturnedParameter area are then copied to the ParameterInput area. The send buffer is also synchronized with the returned parameter values. Automatic synchronization is then always performed when new data is received from the partner object (Par12D_R). If you do not require the parameter, simply leave it open. The default value FALSE then applies; in other words no automatic synchronization is performed.

Name:	**ParameterInput**
Declaration:	INPUT
Data type	ANY
Default:	P#P 0.0 VOID 0 (null pointer)
Explanation	Parameter input area.

Range of values:	P#DBxx.DBX yy.0 DWORD zz xx : Data block number 1...32767 yy : Byte number zz : Number of double words 1...12 starting at byte number yy Example: `P#DB20.DBX 100.0 DWORD 4` Remember the periods and spaces when entering the pointer! No parameter specified: Default (null pointer) is valid. This is, however, not permitted! A pointer >< null pointer must be specified.

The ANY pointer defines the data area in which the parameter values to be acquired are located. This data area must be within a data block and its length can vary between 1 and 12 data double words.

The content of each double word may be a value in double word format (e.g. DINT, REAL etc.); it can also be a mixture of other formats which together form a double word, for example,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

If the parameter setting is incorrect (null pointer, length greater than 12, data area not a data block), an error message to this effect is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

How the parameters at ParameterInput are processed depends on whether they are "hardware" or "software" entries and how the transmission of this data area is triggered. For more information refer to the detailed description of in Function.

Note

When only changed data is transmitted and the data area contains values in double word format, the user is responsible for ensuring that these double word values are actually located in one of the maximum 12 double words of the data area to be acquired. Distribution over two consecutive data double words could otherwise lead to the transmission of only one word of the double word value (high or low word) because a change has occurred in only that particular word. The missing word could lead to processing problems on the partner that receives this value.

Name:	**ReturnedParameter**
Declaration:	INPUT
Data type	ANY
Default:	P#P 0.0 VOID 0 (null pointer)

Explanation Parameter output area.

Range of values:
- P#DBxx.DBX yy.0 DWORD zz
- xx : Data block number 1...32767
- yy : Byte number
- zz : Number of double words 1...12 starting at byte number yy
- Example: `P#DB20.DBX 100.0 DWORD 4`
- Remember the periods and spaces when entering the pointer!
- No parameter specified: Default (null pointer) is valid.

The partner object receiving the parameter values reports back the currently valid local parameter values. These values are displayed at the ReturnedParameter output. If the partner object is set to 'local' and if an input is made there, then parameters changed locally are displayed here at ReturnedParameter.

The ANY pointer defines the data area in which the received parameter values are output. This data area must be within a data block and its length can vary between 1 and 12 data double words. The length must be identical with the length set for ParameterInput.

After startup of the local or partner PLC, or after restoring a connection, an automatic general request ensures that the current, local, valid parameters are displayed at ReturnedParameter.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

If you do not require the parameter, simply leave it open.

If the parameter setting is incorrect (data area not a data block, length greater than 12 or length different from the length set for ParameterInput), an error message to this effect is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

Name: **LocalOperation**
Declaration: OUTPUT
Data type BOOL
Default: FALSE
Explanation Return message from partner object: Object is set to local operation.

Range of values:	Output	Q 0.0 ... Q n.7
	Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7
	No parameter specified: Default value FALSE is valid.	

A parameter can also be set locally at the partner object that receives the parameter. The partner object then must be set to ’local’ at the Local input parameter (see FB Par12D_R below). The current status of the Local input parameter is reported by the partner object and displayed here at the LocalOperation output. As long as the partner object is set to 'local', no parameters are accepted there from other locations.

After startup of the local or partner PLC, or after restoring a connection, an automatic general request ensures that the current, local, valid status is displayed at LocalOperation.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

If you do not require the parameter, simply leave it open.

Name: **NewData**

Refer to Parameters you require often (Page 223).

Explanation of the typical-specific outputs	Whenever the FB has received new data and has output it to the outputs ReturnedParameter or LocalOperation, the NewData output is set to TRUE for one OB1 cycle.

Name: **Release**

Declaration:	IN_OUT
Data type	BOOL
Default:	FALSE
Explanation	Trigger input for sending the currently pending parameter values

Range of values:	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7
	No parameter specified: Default value FALSE is valid.	

You can use this input parameter when the parameter is entered by software, for example at an operator panel (OP). The Release input should then be set using a function key on the OP. You can then enter several parameters initially on the OP. They are only transmitted when the Release function key is activated because the change check only begins with a 1 signal at the Release input and the data double words that have changed since the last transmission are transmitted from the data area defined by ParameterInput.

If you always want to transmit the entire data area defined with ParameterInput and not only the changed parameter values, you should use the RetransmitAll input parameter instead of Release.

The Release input is reset automatically. You should therefore only specify memory or data inputs as the input. The automatic reset would not work with a digital input.

Data checks:

The same safety checks are carried out as with ContinuousEnterFunct = TRUE. Refer to the description there.

If you do not require the parameter, simply leave it open.

Note

The changed data area that is transmitted consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

Example:
The area to be read is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word inclusive.

Note

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

Name:	**RetransmitAll**
Declaration:	IN_OUT
Data type	BOOL
Default:	FALSE
Explanation	Trigger input for transmitting (or retransmitting) the entire data area defined by ParameterInput.

Range of values:	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7
	No parameter specified: Default value FALSE is valid.	

You can use this input parameter when the parameter is entered by software, for example at an operator panel (OP). The RetransmitAll input should then be set using a function key on the OP. When a 1 signal is detected at the RetransmitAll input, the entire data area defined by ParameterInput is transmitted. Changes are not checked.

The RetransmitAll input is reset automatically. You should therefore only specify memory or data inputs as the input. The automatic reset would not work with a digital input. Since there is no change check, this would lead to continuous transmission of all parameter values as long as the digital input has a 1 signal.

The RetransmitAll input can also be used as an option in addition to Release or ContinuousEnterFunct = TRUE when new parameter values were entered but could not be transmitted to the partner (for example because of a disrupted connection or because the partner object was previously set to 'local'). You can then trigger transmission of the entire data area defined by ParameterInput using the RetransmitAll input. All changes that were previously entered but are not yet available at the partner are consistently included.

The RetransmitAll input can also be used as an independent transmission trigger when you always want to send all entries and not just those that have changed. You should then use RetransmitAll instead of Release that only sends the changed parameter values.

Data checks:

The same safety checks are carried out as with ContinuousEnterFunct = TRUE. Refer to the description there.

If you do not require the parameter, simply leave it open.

Note

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

Name:	**ApplyRemoteParamMan**
Declaration:	IN_OUT
Data type	BOOL
Default:	FALSE
Explanation	Trigger input for synchronization of the input area with the returned area.

Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid.

The input triggers a one-time synchronization of the ParameterInput input area with the ReturnedParameter area. All the parameter values from the ReturnedParameter area are then copied to the ParameterInput area. The send buffer is also synchronized with the returned parameter values.

The ApplyRemoteParamMan input is reset automatically. You should therefore only specify memory or data inputs as the input. The automatic reset would not work with a digital input. The result would be a constant synchronization as long as the digital input has a 1 signal.

If you do not require the parameter, simply leave it open.

Note

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

2.5.13 ST7 parameter typical FB Par12D_R

Function

Receive 1 to 12 parameter values (each 1 double word) or enter locally and send back the current, locally valid parameter values.

The content of each double word may be a value in double word format (e.g. DINT, REAL etc.); it can also be a mixture of other formats which together form a double word, for example,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

The data area in which the received parameter values are output is defined with the ParameterOutput parameter in the form of an Any pointer. This data area must be within a data block and its length can vary between 1 and 12 double words. You can also use the block to enter the parameter values locally. The input area for this is defined as an Any pointer with the LocalParameterInput parameter. It must be located within a data block and its length must be identical to the length configured at the ParameterOutput parameter.

The block only sends the changed data area. However, the complete parameter set is returned in response to a general or single request.

Bumpless switchover between the Local and Remote operating modes is guaranteed.

Explanation of the parameters

Names:	**PartnerNo**
	PartnerObjectNo
	Enabled
	ImageMemory
	Conditional
	Unconditional
	TimeStamp

Refer to Parameters you require often (Page 223).

Name:	**Local**		
Declaration:	INPUT		
Data type	BOOL		
Default:	FALSE		
Explanation	Local parameter input released.		
	Range of values:	TRUE or FALSE	
		Input	I 0.0 ... I n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified: Default value FALSE is valid	

This input is used to enable local input of a parameter over the data area addressed with LocalParameterInput. Parameters sent for example by the master are not accepted by the object as long as Local = TRUE.

The current status of the Local input is transferred to the partner.

Bumpless switchover:

- When there is a switchover from Local = 0 to Local = 1, the last values at ParameterOutput are held until new parameter values are entered over LocalParameterInput.
- When there is a switchback from Local = 1 to Local = 0, the last values at ParameterOutput are held until the block receives new parameter values from the remote partner.

Special case:
You can also enter the parameter values during local input directly in the output area defined by ParameterOutput. Either you do not specify an input area for LocalParameterInput or you specify the same data area both for LocalParameterInput and ParameterOutput.
This type of the parameter entry cannot be prevented by the Local input. Regardless of the Local status, the values entered in the output area are sent immediately to the partner by the function block.
Local parameter entries can therefore be made regardless of the status of the Local input. Local only influences the acceptance of parameters sent by the remote partner:

- If Local = 0, the parameters sent by the remote partner are accepted and output to the ParameterOutput data area.
- If Local = 1, any parameters sent by the remote partner are rejected.

In this special situation, the Release input and ContinuousEnterFunct have no function.

A status change of the Local parameter is always sent by the TIM according to the send buffer principle (even when the parameter ImageMemory = TRUE). This ensures that the optional synchronization of the input and output area on the partner is always performed correctly (see FB Par12D_S, parameters ApplyRemoteParamMan and ApplyRemoteParamAuto).

Note

Please read the note on the ContinuousEnterFunct parameter.

Name:	**ContinuousEnterFunct**
Declaration:	INPUT
Data type	BOOL
Default:	FALSE
Explanation	Continuous local parameter acquisition.

Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid.

With this parameter, you can decide whether the parameter values in the LocalParameterInput input area should be continuously read in and changes checked. The change check is implemented by comparing the current parameter values at ParameterOutput. Changes in the input area are copied immediately to the output area and transmitted to the partner. Only changed values are sent. If there is more than one change, the block sends the data area in which all changed parameter values are located.

The ContinuousEnterFunct = TRUE parameter setting only takes effect when the following conditions are met:

- An input area is defined by the LocalParameterInput parameter and this is not identical to the output area defined by ParameterOutput.
  and
- There is a 1 signal at the Local input (= TRUE).

This method of local parameter acquisition is suitable when the parameter values are entered in the LocalParameterInput area by software, but can also be used for entering the parameters from an operator panel (OP) when the OP has no separate function key with which to trigger acceptance. For more detailed information and related parameters, refer to the section Function.

If you do not require the parameter, simply leave it open.

Note

The changed data area that is transmitted consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

Example:
The area to be read is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word inclusive.

Note

When ContinuousEnterFunct = TRUE, the values available at LocalparameterInput are entered immediately and passed to ParameterOutput when the 1 signal at the Local input is detected assuming the local input values differ from the current parameter value output at this point in time.

Name:	**LocalParameterInput**
Declaration:	INPUT
Data type	ANY
Default:	P#P 0.0 VOID 0 (null pointer)
Explanation	Local parameter input area.

Range of values:	P#DBxx.DBX yy.0 DWORD zz xx : Data block number 1...32767 yy : Byte number zz : Number of double words 1...12 starting at byte number yy Example: `P#DB20.DBX 100.0 DWORD 4` Remember the periods and spaces when entering the pointer! No parameter specified: Default (null pointer) is valid.

The ANY pointer defines the data area in which the parameter values to be acquired locally are located. This data area must be within a data block and its length can vary between 1 and 12 data double words. The length must be identical with the length set for ParameterOutput.

The content of each double word may be a value in double word format (e.g. DINT, REAL etc.); it can also be a mixture of other formats which together form a double word, for example,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

If you do not require the parameter, simply leave it open.

If the parameter setting is incorrect (data area not a data block, length greater than 12 or length different from the length set for ParameterOutput), an error message to this effect is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

Note

Only changed data is transmitted to the partner. If the data area contains values in double word format, the user is responsible for ensuring that these double word values are actually located in one of the maximum 12 double words of the data area to be acquired. Distribution over two consecutive data double words could otherwise lead to the transmission of only one word of the double word value (high or low word) because a change has occurred in only that particular word. The missing word could lead to problems in processing on the partner that receives this value (applies, for example to ST7cc, but not to an S7 CPU).

Name:	**ParameterOutput**
Declaration:	INPUT
Data type	ANY
Default:	P#P 0.0 VOID 0 (null pointer)
Explanation	Parameter output area.

Range of values:

P#DBxx.DBX yy.0 DWORD zz

xx : Data block number 1...32767

yy : Byte number

zz : Number of double words 1...12 starting at byte number yy

Example: `P#DB20.DBX 100.0 DWORD 4`
Remember the periods and spaces when entering the pointer!

No parameter specified: Default (null pointer) is valid. This is, however, not permitted! A pointer >< null pointer must be specified.

The ANY pointer defines the data area in which the locally entered parameter values or those received from the partner are output. This data area must be within a data block and its length can vary between 1 and 12 double words.

The content of each double word may be a value in double word format (e.g. DINT, REAL etc.); it can also be a mixture of other formats which together form a double word, for example,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

FB Par12D_R stores the received data without further processing in the data area defined by ParameterOutput. The user program is responsible for evaluating and processing received data.

When only changed data is sent by the partner object Par12D_S, it is possible that only part of the data output area is newly written, namely, the area in which the changes were detected at the acquisition end.

If the parameter setting is incorrect (null pointer, length greater than 12, data area not a data block), an error message to this effect is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

Note

When only the changed data area is received, this area consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

Example:
The area to be read is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word inclusive.

Name: **NewData**

Refer to Parameters you require often (Page 223).

Explanation of the typical-specific outputs: Whenever the FB has received new parameter values from the partner object and has output them to the output field ParameterOutput, the NewData output is set to TRUE for one OB1 cycle. This also applies when there is new local input when Local = 1.

Name: **Release**

Declaration: IN_OUT

Data type: BOOL

Default: FALSE

Explanation: Input for the acceptance of local parameter entry.
No parameter specified: Default value FALSE is valid

Range of values:		
	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

The acceptance of the parameter value at the LocalParameterInput parameter input can be triggered over this input by a signal edge change.

A change from 0 to 1 at the Release input is taken into account only when the following conditions are met:

- An input area is defined by the LocalParameterInput parameter and this is not identical to the output area defined by ParameterOutput.
  and
- There is a 1 signal at the Local input (= TRUE).

You can use this Release input parameter when the parameter is entered by software, for example at an operator panel (OP). The Release input should then be set using a function key on the OP. You can then enter several parameters initially on the OP. The parameter values are read in and checked for changes only when the Release function key is activated. The change check is implemented by comparing the current parameter values at ParameterOutput. Changes in the input area are then copied immediately to the output area and transmitted to the partner. Only changed values are sent. If there is more than one change, the block sends the data area in which all changed parameter values are located.

The Release input is reset automatically. Instead of a memory bit or data bit, a digital input can also be specified as the input. The automatic reset would not work with a digital input. This does not, however, have negative effects. The triggering of the acquisition over Release is triggered by a signal edge change; in other words, it occurs only once.

If you do not require the parameter, simply leave it open.

Note

The changed data area that is transmitted consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

Example:
The area to be read is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word inclusive.

Note

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

2.5.14 ST7 data typical FB Dat12D_S

Function

Send maximum of 12 double words with any data content.

The content of each double word may be a value in double word format (e.g. DINT, REAL etc.); it can also be a mixture of other formats which together form a double word, for example,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

Sending the data area can be triggered in two ways:

- By a change check. Transmission is triggered as soon as a bit changes. (Parameter SendOnChange = TRUE)
- The user program decides when a transmission will take place (signal edge change from 0 to 1 at TriggerInput).
  This could also be a time-driven transmission. This case, you could use FC Trigger.

You can also specify whether the transmission includes all data or only the data double words that have changed.

Note

When only changed data is transmitted and the data area contains values in double word format, the user is responsible for ensuring that these double word values are actually located in one of the maximum 12 double words of the data area to be acquired. Distribution over two consecutive data double words could otherwise lead to the transmission of only one word of the double word value (high or low word) because a change has occurred in only that particular word. The missing word could lead to problems in processing on the partner that receives this value (applies, for example to ST7cc, but not to an S7 CPU).

Explanation of the parameters

Names: **PartnerNo**
PartnerObjectNo
Enabled
ImageMemory
Conditional
Unconditional
TimeStamp

Refer to Parameters you require often (Page 223).

Name: **SendOnChange**
Declaration: INPUT
Data type BOOL
Default: FALSE
Explanation Send on change.

Range of values: TRUE or FALSE
No parameter specified: Default value FALSE is valid.

Here, you specify whether the FB checks for changes within the acquired data area (to determine whether at least one bit has changed). If a change is detected, a transmission of the data area is triggered automatically. Whether the entire area is transmitted or only the changed part can be specified with the SendAll parameter (refer to the explanations on the relevant parameter).

If no parameter is specified, the default is FALSE; in other words, there is no change-driven data transmission. Transmission must then be triggered by the user at the TriggerInput input parameter.

Name: **TriggerInput**
Declaration: INPUT

Data type	BOOL		
Default:	FALSE		
Explanation	Trigger input.		
	Range of values:	Input	I 0.0 ... I n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
		No parameter specified: Default value FALSE is valid.	

If required, this parameter can be used to specify an input over which the user can trigger the transmission *) of the data message at any time (signal edge from 0 to 1).

Example:
Time-driven analog value transmission with time stamp for supplying an analog value archive in the control center. Note: To prevent these messages with time stamps from being overwritten when saving on the station TIM, the ImageMemory parameter must be set to FALSE.

FC Trigger can be used for time-driven triggering of a transmission over TriggerInput (for more detailed information, refer to the description of this block).

If you do not require the parameter, simply leave it open. You should, however, then set the SendOnChange parameter to TRUE so that the data is transmitted automatically at every change.

*) TriggerInput actually only triggers transmission indirectly. With a 0/1 edge at TriggerInput, the message is put together with its current values and transferred to the local TIM. The TIM is responsible for the actual transmission to the partner. Transmission is immediate over a dedicated line or wireless link; with a dial-up connection, it is possible that the message is saved first on the TIM and sent at a later point in time (for example, because the message is marked as a "conditional spontaneous" message; see the Conditional parameter).

Note

You can also select a combination of SendOnChange plus TriggerInput. This means that a transmission is triggered both when a change is detected and at every signal edge change from 0 to 1 at the TriggerInput.

Note

If you use neither SendOnChange nor TriggerInput to trigger data transmission, the data will only be transmitted when there is a single request for this data object or within the framework of a general request.

Name:	**SendAll**
Declaration:	INPUT

Data type	BOOL
Default:	TRUE
Explanation	Send all data with every transmission.
	Range of values: TRUE or FALSE
	Here, you specify whether the FB will transfer or data of the area defined with DataInput or only changed data. The transmission can be triggered by the activated change check (SendOnChange = TRUE) or by TriggerInput.
	• SendAll = TRUE always send all data • SendAll = FALSE send only changed data – Exception: If a transmission is triggered over TriggerInput, and no data has changed at this point in time, the entire area is transmitted (in this exceptional situations, corresponds to SendAll = TRUE).
	If no parameter is specified, the default TRUE applies; in other words, the entire area is always transmitted.

Note

When only the changed data area is transmitted (SendAll = FALSE), this area consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

Example:
The area to be read is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word inclusive.

Note

If there is a single request for this data object or within the framework of a general request, all data words of the area defined by DataInput are always transmitted.

Name:	**DataInput**
Declaration:	INPUT
Data type	ANY
Default:	P#P 0.0 VOID 0 (null pointer)
Explanation	Data input area.

Range of values:	P#DBxx.DBX yy.0 DWORD zz xx : Data block number 1...32767 yy : Byte number zz : Number of double words 1...12 starting at byte number yy Example: `P#DB20.DBX 100.0 DWORD 4` Remember the periods and spaces when entering the pointer! No parameter specified: Default (null pointer) is valid. This is, however, not permitted! A pointer >< null pointer must be specified.

The ANY pointer describes the data area in which the data to be acquired is located. This data area must be within a data block and its length can vary between 1 and 12 data double words.

The content of each double word may be a value in double word format (e.g. DINT, REAL etc.); it can also be a mixture of other formats which together form a double word, for example,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

If the parameter setting is incorrect (null pointer, length greater than 12, data area not a data block), an error message to this effect is entered in the diagnostics buffer (event ID B114, [Info2/3] = 11). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

Note

When only changed data is transmitted and the data area contains values in double word format, the user is responsible for ensuring that these double word values are actually located in one of the maximum 12 double words of the data area to be acquired. Distribution over two consecutive data double words could otherwise lead to the transmission of only one word of the double word value (high or low word) because a change has occurred in only that particular word. The missing word could lead to problems in processing on the partner that receives this value (applies, for example to ST7cc, but not to an S7 CPU).

2.5.15 ST7 data typical FB Dat12D_R

Function

Receive maximum of 12 double words with any data content.

The content of each double word may be a value in double word format (e.g. DINT, REAL etc.); it can also be a mixture of other formats which together form a double word, for example,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

FB Dat12D_R stores the received data without further processing in the data area defined by DataOutput. The user program is responsible for evaluating and processing received data.

Explanation of the parameters

Names:	**PartnerNo**
	PartnerObjectNo
	Enabled

Refer to Parameters you require often (Page 223).

Name:	**DataOutput**
Declaration:	INPUT
Data type	ANY
Default:	P#P 0.0 VOID 0 (null pointer)
Explanation	Data output area.

Range of values:	P#DBxx.DBX yy.0 DWORD zz xx : Data block number 1...32767 yy : Byte number zz : Number of double words 1...12 starting at byte number yy Example: P#DB20.DBX 100.0 DWORD 4 Remember the periods and spaces when entering the pointer! No parameter specified: Default (null pointer) is valid. This is, however, not permitted! A pointer >< null pointer must be specified.

The ANY pointer describes the data area in which the received data will be stored. This data area must be within a data block and its length can vary between 1 and 12 data double words.

The content of each double word may be a value in double word format (e.g. DINT, REAL etc.); it can also be a mixture of other formats which together form a double word, for example,

- 4 bytes, or
- 2 words, or
- 2 bytes plus 1 word.

FB Dat12D_R stores the received data without further processing in the data area defined by DataOutput. The user program is responsible for evaluating and processing received data.

When only changed data is sent by the partner object Dat12D_S, it is possible that only part of the data output area is newly written, namely, the area in which the changes were detected at the acquisition end.

How to read out the time stamp received with the data is described in the section Notes on the SINAUT time stamp.

If the parameter setting is incorrect (null pointer, length greater than 12, data area not a data block), an error message to this effect is entered in the diagnostics buffer (event ID B114, [Info2/3] = 4). The CPU does not change to STOP. The FB is then no longer processed, however, until the parameter assignment error has been corrected.

Note

When only the changed data area is received, this area consists of the first and the last double word in which a change was detected and all words located in between, even if these have not changed.

Example:
The area to be read is 10 double words long. In this case, changes were detected in the 2nd, 5th and 8th double words. The transmitted area is therefore from the 2nd to the 8th double word inclusive.

Name:	**NewData**
Refer to Parameters you require often (Page 223).	
Explanation of the typical-specific outputs	Whenever the FB has received new data and has stored it in the data area defined by DataOutput, the NewData output is set to TRUE for one OB1 cycle.

2.6 Blocks for optional expansion

2.6.1 FC ListGenerator300, FC ListGenerator400

Function

The FC ListGenerator is required in a CPU that receives messages containing either an incomplete destination address or no destination address at all. The lack of the destination object number is the most important factor here because this points to the instance DB in which the received information should be stored. Missing or incomplete destination addresses can occur when no or incomplete parameters are set for them in the station (this is permitted for typicals that send binary information, analog values or counted values). If these typicals send data to more than one destination, no destination address is set for these typicals. Due to the missing destination information, the send message is automatically transmitted to all destinations for which a connection is configured. Such messages are therefore received without a destination address at the various destinations.

Note

Messages without a destination address that are sent to the partner over a TIM have the destination subscriber number added by the sending TIM, and sometimes several addresses if there are several destinations along the way. The TIM enters 0 in the address field for the destination object number, since the TIM does not have the relevant information. The only destination subscribers it knows are those to which it has a configured connection. At the receiving end, the message therefore contains the destination subscriber number but the destination object number is 0.

If the destination object number is not contained in the received message, FC Distribute, which is responsible for distributing the received messages, references an object reference list. Using the source address (source subscriber no. + source object no.) contained in every message, FC Distribute searches through the list for an entry that specifies the missing destination object number for the given source address; in other words, it searches for the number of the local instance DB.

This object reference list is created by FC ListGenerator. The FC has no parameters. It must be included in the cyclic SINAUT program (in OB1) following FC BasicTask.

Note

There is an FC ListGenerator version for S7-300 (symbolic name, ListGenerator300) and for S7-400 (symbolic name ListGenerator400).

When creating the list, FC ListGenerator uses the addresses set in the parameters for the typicals that receive data. The parameters PartnerNo and PartnerObjectNo are mandatory for these typicals. These parameters are identical with the source address in the corresponding received message. Since the typical also knows the number of its instance DB, it therefore knows all the addresses required for an entry in the reference list. During startup, FC ListGenerator makes a request to all typicals that can receive data to enter their configured addresses with the number of the instance DB in the reference list. The object reference list therefore does not require special parameter settings, it is simply created from the existing parameters of the receive typicals and is therefore always consistent.

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

How it works

FC ListGenerator creates the list(s) after startup in three consecutive OB1 cycles:

1. In the first cycle, it determines how many entries will be required in the first and, if applicable, in the second object reference list. The typicals involved only increment a counter during this run.
2. In the second cycle, FC ListGenerator generates the data block for the first and, if applicable, the second object reference list with the required length and enters 0 in all the data words. During the same cycle, each typical involved enters its addresses and the number of the corresponding instance DB in the list.
3. In the third and final cycle, FC ListGenerator sorts all the entries in ascending order. Sorting speeds up the search in the list during actual operation.

When generating the data block, FC ListGenerator does the following:

If a list has not yet been created, it first searches for a free DB number; the first free DB number below the number for DB BasicData is used.

If a list already exists, FC ListGenerator checks to see if its present length is adequate for the currently required number of references. If the length is adequate, 0 is entered as the content and the addresses are written again and sorted.

If the existing data block is too short, different procedures are used for S7-300 and S7-400:

- For S7-300, a new DB is generated. The old DB remains in memory because S7- 300 has no delete function for data blocks. This DB must be deleted by the user with the programming device. If there is not enough memory on the CPU to be able to generate a new DB, the existing DB must be deleted by the user before restarting.
- In an S7-400, the existing DB is deleted and a new DB is generated with the same number and the new length.

2.6.2 FC TimeTask

Function

FC TimeTask is responsible for continuous date and time management on a CPU.

The FC has no parameters. It must be included in the cyclic SINAUT program (in OB1) following FC BasicTask.

FC TimeTask requires that the CPU is synchronized by a local TIM. This synchronization must be enabled in the 'Time Services' tab in the Properties dialog for the corresponding TIM. See figure below.

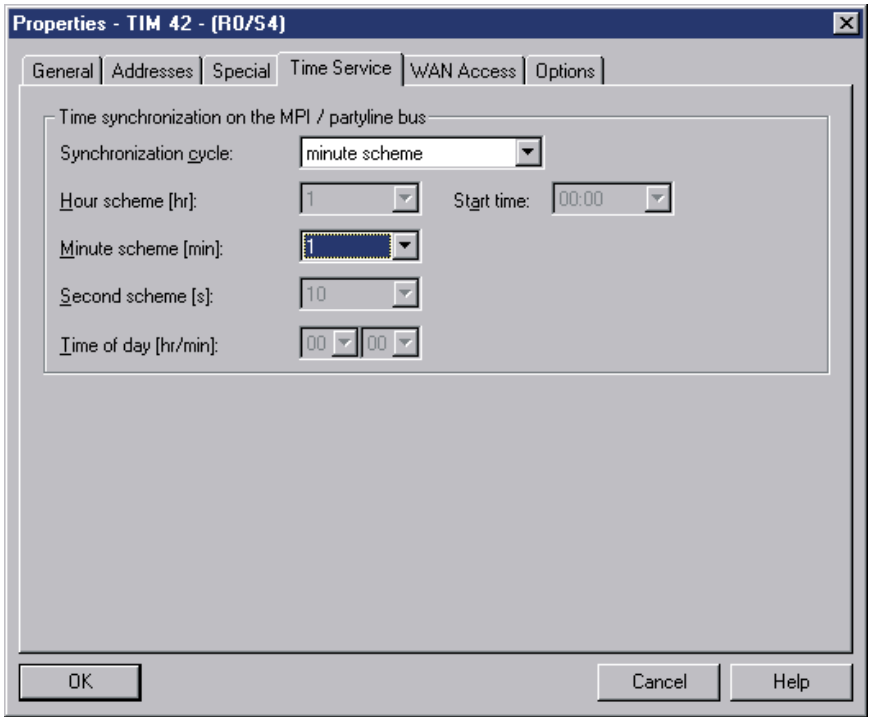


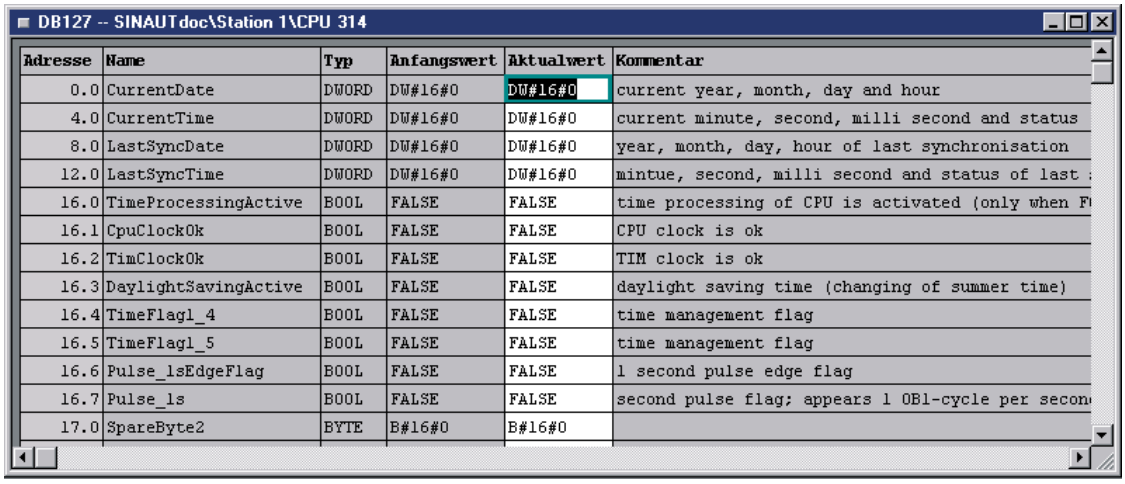
Figure 2-27 TIM parameter assignment tabs - setting the synchronization on the MPI / party line

For more detailed information on setting the synchronization, refer to the chapter "Configuration software for SINAUT ST7" in this manual, section 'Parameter assignment for TIM modules'.

After the CPU has started up, the TIM supplies the current date/time for the first time in an organizational message. Following this, the synchronization continues at the time interval specified by the configuration of the TIM (a one minute time scheme is recommended for synchronization on MPI/party line).

FC TimeTask sets the CPU clock to the synchronized time received from the TIM and reads this clock in every OB1 cycle. As long as it continues to advance and remains plausible, the read time is entered in the first two double words of DB BasicData and is marked as valid or invalid and as daylight-saving or standard time. All blocks take the current time from there as they need it. For example, data point typicals do this to time stamp their messages, as does

FC Trigger to check whether a point in time configured for the FC has been reached or a specified time period has elapsed. This time is, of course, also available to the user program.



DB127 -- SINAUTdoc\Station 1\CPU 314

Adresse	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	CurrentDate	DWORD	DW#16#0	DW#16#0	current year, month, day and hour
4.0	CurrentTime	DWORD	DW#16#0	DW#16#0	current minute, second, milli second and status
8.0	LastSyncDate	DWORD	DW#16#0	DW#16#0	year, month, day, hour of last synchronisation
12.0	LastSyncTime	DWORD	DW#16#0	DW#16#0	mintue, second, milli second and status of last
16.0	TimeProcessingActive	BOOL	FALSE	FALSE	time processing of CPU is activated (only when F
16.1	CpuClockOk	BOOL	FALSE	FALSE	CPU clock is ok
16.2	TimClockOk	BOOL	FALSE	FALSE	TIM clock is ok
16.3	DaylightSavingActive	BOOL	FALSE	FALSE	daylight saving time (changing of summer time)
16.4	TimeFlag1_4	BOOL	FALSE	FALSE	time management flag
16.5	TimeFlag1_5	BOOL	FALSE	FALSE	time management flag
16.6	Pulse_1sEdgeFlag	BOOL	FALSE	FALSE	1 second pulse edge flag
16.7	Pulse_1s	BOOL	FALSE	FALSE	second pulse flag; appears 1 OB1-cycle per secon
17.0	SpareByte2	BYTE	B#16#0	B#16#0	

Figure 2-28 DB BasicData, CurrentDate and CurrentTime

Table 2- 5 The exact assignment of the data words with data, time of day and time status:

CurrentDate	Data byte 0	Year * 10	Year * 1
	Data byte 1	Month * 10	Month * 1
	Data byte 2	Day * 10	Day * 1
	Data byte 3	Hour * 10	Hour * 1
CurrentTime	Data byte 4	Minute * 10	Minute * 1
	Data byte 5	Second * 10	Second * 1
	Data byte 6	Millisecond * 100	Millisecond * 10
	Data byte 7	Millisecond * 1	Time status

Table 2- 6 Assignment of the 4 time status bits:

2^0	0 = Date/time invalid 1 = Date/time valid
2^1	0 = Standard time 1 = Daylight saving time
2^2	(not used)
2^3	(not used)

In addition to using the time status, data bit 16.1 *CpuClockOk* also indicates whether the date/time is valid. Once the time of day is valid on the CPU, this bit is set to 1 by FC TimeTask. In the user program this bit can be directly queried using its symbolic name *"BasicData".CpuClockOk*.

2.6.3 FC Trigger

Function

This function block sets an output (memory bit, data bit or digital output) at a point in time or at time intervals specified by the user. The block resets this output after one OB1 cycle.

The FC can be called from any point and also several times in the cyclic SINAUT program (in OB1).

When an individual program section or software function is to be triggered using FC Trigger, it is advisable to call it directly before the block that, for example, will be executed conditionally due to the memory bit set by FC Trigger, or will execute a certain function triggered by the memory bit (e.g. triggering a counted value transmission every 2 hours).

If you want to activate several (software) functions at the same time, this can be done with one FC Trigger: All query the same memory bit set by FC Trigger. However, this only works without problems when the triggered blocks do not reset this memory bit themselves. To avoid the problem of triggered blocks resetting the memory bit:

- Insert a suitable number of Trigger FCs, each with the same time but with a different output memory bit, or
- Set a suitable number of additional memory bits after FC Trigger has set its output memory bit.

The FC accesses the SINAUT time in the first two data double words in DB BasicData. These are continuously updated if an FC TimeTask is included in the SINAUT program and this is synchronized with the date and time of day by a local TIM at regular intervals. FC Trigger only compares its time parameters with the current time of day if the SINAUT time is marked as O.K. (for a valid time of day bit 0 is set in DB BasicData in data byte 7, the time status byte).

The precision with which FC Trigger operates depends, on the one hand, on the exactness of the SINAUT time and, on the other, on the OB1 cycle time. If the OB1 cycle time is less than 1 s (this is usually the case), the output is set exactly to the set second value (with the inaccuracy of the OB1 cycle time of less than 1 sec.). If the OB1 cycle time is greater than 1 s, the FC works with a tolerance window of 4 s; in other words, if the FC is processed late but still within 4 s of the configured time or time interval, the output is still set.

The edge memory bit (Flag parameter) to be set for the FC is also set with the output and reset 5 s after the specified point in time or time interval. It is not allowed to use a dummy parameter for the edge memory bit or to reset it with the user program!

Following the descriptions of the parameters below, you will find several examples that explain how to set parameters for points in time or time intervals using FC Trigger.

Explanation of the parameters

Name:	**IntervalMode**
Declaration:	INPUT

Data type	BOOL
Explanation	An interval is set for the FC. Range of values: TRUE or FALSE FALSE = no, in other words, a time is set. TRUE = Yes, in other words, an interval is set. You will find examples of setting a point in time or a time interval following the description of the parameters.

Name:	**Hour_Minute**
Declaration:	INPUT
Data type	WORD
Explanation	Specifies the values for hours and minutes. Further explanation: Refer to Month_Year parameter below.

Name:	**Second_Day**
Declaration:	INPUT
Data type	WORD
Explanation	Specifies the values for seconds and day. Further explanation: Refer to Month_Year parameter below.

Name:	**Month_Year**
Declaration:	INPUT
Data type	WORD
Explanation	Specifies the values for month and year. Range of values: 00 ... 99, or FF Each parameter has two parts, each of the two values per parameter is specified with two digits as a BCD coded value. The first two digits contain the values for hours, seconds or month; the two remaining digits specify the minutes, day or year. Enter FF for parameters you do not require. The parameters allowed depend on the particular parameter and the IntervalMode parameter. For more information, refer to examples following the descriptions of these parameters.

Name:	**TriggerOutput**
Declaration:	OUTPUT
Data type	BOOL
Explanation	Trigger output.

Range of values:	Output	Q 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

The output is set for the duration of one OB1 cycle when the point in time or time interval specified by Hour ... Year is reached.

Name:	**Flag**
Declaration:	IN_OUT
Data type	BOOL
Explanation	Edge memory bit for TriggerOutput.

Range of values:	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

No dummy memory bit is permitted. The edge memory bit must also not be reset by the user program.

Note

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter and this should not be used.

Examples of setting a time or time interval

IntervalMode = FALSE (or 0)

The FC works with a point in time. When the set point in time is reached, TriggerOutput is set for one OB1 cycle.

All time parameters can be used to specify the time. If you do not require a parameter, set FF. When checking whether a time has been reached, these are ignored. This can be used in some areas (although IntervalMode = 0) for setting a time scheme. Refer to the following examples.

Permitted ranges for the time parameters:

Hours	00-23	Day	01-31
Minutes	00-59	Month	01-12
Seconds	00-59	Year	00-99

Examples:

1. IntervalMode: FALSE The TriggerOutput output is set once
   Hour_Minute : W#16#0645 at 06:45:12 on 04.02.91.
   Second_Day : W#16#1204
   Month_Year : W#16#0291
2. IntervalMode: FALSE The TriggerOutput output is set every
   Hour_Minute : W#16#0600 day at 06:00:00.

Second_Day : W#16#00FF
Month_Year : W#16#FFFF

3. IntervalMode: FALSE The TriggerOutput output is set on every
   Hour_Minute : W#16#0600 1st of a month at 06:00:00
   Second_Day : W#16#0001.
   Month_Year : W#16#FFFF
4. IntervalMode: FALSE The TriggerOutput output is set every
   Hour_Minute : W#16#0600 year on October 1st
   Second_Day : W#16#0001 at 06:00:00.
   Month_Year : W#16#10FF
   IntervalMode = TRUE (or 1)

The FC works on the time interval principle. When the set point in time value is reached, TriggerOutput is set for one OB1 cycle.

Only the entries for hours, minutes and seconds are relevant. The date parameters are ignored. A time interval can only be set in hours or minutes or seconds. If you do not require a time parameter, set FF.

The following time intervals are permitted:
Hours: 01, 02, 03, 04, 06, 08, 12, 24.
Minutes : 01, 02, 03, 04, 05, 06, 10, 12, 15, 20, 30, 60.
Seconds : 10, 12, 15, 20, 30, 60.

Examples:

1. IntervalMode: TRUE Every 6 hours the output
   Hour_Minute : W#16#06FF TriggerOutput is set at:
   Second_Day : W#16#FFFF 00:00:00, 06:00:00, 12:00:00 and
   Month_Year : W#16#FFFF 18:00:00.
2. IntervalMode: TRUE Every 30 minutes the output
   Hour_Minute : W#16#FF30 TriggerOutput is set at:
   Second_Day : W#16#FFFF 00:00:00, 00:30:00, 01:00:00,
   Month_Year : W#16#FFFF 01:30:00, 02:00:00, 02:30:00 etc.

Error message during startup

The FC checks the parameters Hour_Minute, Second_Day and Month_Year every cycle to make sure that the permitted range of values is kept to. What is permitted depends to some extent on the IntervalMode parameter. If the parameter setting is incorrect, an error message to this effect is entered in the diagnostic buffer only during startup (event ID B113). The CPU does not change to STOP. The FC then checks only the parameters (without an error message) until the parameter assignment error is cleared.

The diagnostic message provides an exact description of the bad parameter (consecutive number of the parameter, i.e. 2, 3 or 4). Depending on the parameter IntervalMode, the diagnostics message may be caused by the following situation:
IntervalMode = FALSE (or 0)

The range permitted for the parameters hours, minutes, seconds, day, month, year has been violated. Apart from FF, the following settings are valid:

Hours	00-23	Day	01-31
Minutes	00-59	Month	01-12
Seconds	00-59	Year	00-99

IntervalMode = TRUE (or 1)

In this case, there are two possible causes for an error:

1. The range permitted for the parameters hours, minutes, and seconds was violated. Apart from FF, the following settings are valid:
   Hours : 01, 02, 03, 04, 06, 08, 12, 24.
   Minutes : 01, 02, 03, 04, 05, 06, 10, 12, 15, 20, 30, 60.
   Seconds : 10, 12, 15, 20, 30, 60.
2. A time interval can only be set in hours or minutes or seconds. The two unused parameters must be specified as FF. An error also occurs when all three parameters are set to FF.

2.6.4 FC PulseCounter

Function

The FC PulseCounter is responsible for counted pulse acquisition.

A maximum of 8 pulse trains are acquired over digital inputs and using SIMATIC counters passed to the function blocks which create the counted value messages (Cnt01D_S, Cnt04D_S and ZTZ01, ZTZ02, ZTZ03).

The acquisition of the counted pulses is time-driven. To this purpose, the FC PulseCounter is included in a cyclic interrupt OB, for example, OB35. The call interval for the cyclic interrupt OB must be matched to the pulse duration of the counted pulses. For more information on the cyclic interrupt OB, refer to the section ’Time-driven SINAUT program in a cyclic interrupt OB’.

Explanation of the parameters

Name:	**InByte**		
Declaration:	INPUT		
Data type	BYTE		
Explanation	Input byte for counted pulses.		
	Range of values:	Input bytes	PIB0 ... PIBn
		Memory bytes	MB0 ... MBn LB0 ... LBn
		Data bytes	DBm.DBB0 ... n
	The inputs for counted pulse acquisition can be set byte by byte.		

Note

If an input byte of a digital input is specified, it must be the address of the I/O byte (PIB) direct from the digital input modules. The current status of the counted value input can only be reliably acquired by direct access. Counted pulses may go undetected when read from the process image of the inputs (IB).

Name:	**EnableMask**
Declaration:	INPUT
Data type	BYTE
Explanation	Enable mask for the counted value inputs.
	Range of values: B#16#00 to B#16#FF
	EnableInMask provides a bit mask for specifying the inputs in the input byte to which counted pulses are actually connected. The following applies to each bit in the bit mask:
	0 = Input bit for acquisition disabled 1 = Input bit for acquisition enabled

Note

The mask can only be edited in hexadecimal format B#16#00 to B#16#FF. Entry as 8-bit binary numbers from 2#0 to 2#1111 1111 is not possible for the data type BYTE.

The assignment of the bits in the mask to the inputs in the InByte input byte:

	InByte	.7	.6	.5	.4	.3	.2	.1	.0
EnableMask	B#16#	0 … F					0 … F		

Example: EnableInMask : B#16#83
The following are enabled: Inputs .7, .1 and .0
The following are disabled: Inputs .6 to .2

Name:	**CntIn_0 ... CntIn_7**
Declaration:	INPUT
Data type	COUNTER
Explanation	Pulse counter
	Range of values: C0 or C1 ... Cn (n is CPU-dependent)

For each enabled counted value input, a SIMATIC counter must be specified with the corresponding parameter CntIn_0 ... CntIn_7. The SIMATIC counter is incremented with each acquired pulse.

The counters configured here must be specified for the actual counted value function blocks, Cnt01D_S, Cnt04D_S and ZTZ01, ZTZ02, ZTZ03 as input counters (parameter Counter_1 ... _4). These function blocks read out the assigned counter and then reset it.

Counter C0 is recommended as a placeholder for parameters that are not required.

Example of the setting for CntIn_0 ... CntIn_7 based on EnableInMask : = B#16#83

CntIn_0 : = C10
CntIn_1 : = C11
CntIn_2 : = C0
CntIn_3 : = C0
CntIn_4 : = C0
CntIn_5 : = C0
CntIn_6 : = C0
CntIn_7 : = C12

2.6.5 FC Safe

Function

The block ensures reliable entry of commands and setpoints. When an entry is pending, the FC checks to determine whether only one entry is waiting in the current OB1 cycle and then enables the reading block.

FC Safe should be called in the cyclic SINAUT program (in OB1) in every CPU in which commands and/or setpoints are acquired once at the end of all command and setpoint FBs. For more detailed information on the program structure, see the section ’The cyclic OB1 program for a control center’.

The FC has separate monitoring functions for commands and setpoints entered by the hardware (input modules) or those entered by software (operator panel (OP) etc.). These two entry methods can be used at the same time. The FC then enables input from the hardware or software 'track' separately, and when necessary even simultaneously. As a basic rule, however, only one command or setpoint input may be detected per ’track’.

For hardware input there is an additional condition: The input must remain constant for specific period of time. This delay time is set for FC Safe with the InputDelayTime parameter. The input is released after the delay time has elapsed only if the currently entered command or setpoint remains unchanged for the specified delay time and no other command or setpoint input is detected during this time. The actual creation of the command or setpoint message is handled by the block that read in the command or setpoint.

FC Safe provides two other code bits for hardware input:

InputOK :	Has signal 1 as soon as the current entry is enabled. The code bit goes off when the entry is reset, i.e. as soon as the command key is released or, for setpoint input, as soon as the key at the EnterInput input is released.
InputError :	Has signal 1 as soon as an input error within the hardware 'track' is detected. Either two or more simultaneous command or setpoint entries have been detected or at one of the inputs the 1 signal was detected over a long period if time, i.e. the input is 'stuck'. This long monitoring time can be configured for FC Safe with the MaxInputTime parameter.

FC Safe indicates a command output error recognized in a station using the GlobalCmdOutputError output. A command output error can only occur at the receiving end either when the content of both command bytes in the received message is not identical or when more than one bit is set in the command byte. When such an error is detected, the station reports this back with an organizational message to the subscriber that sent the command message. FC Safe on this subscriber then indicates the error at the GlobalCmdOutputError output. This is a group display. When an error is detected the output remains set to signal 1 until the user resets the group signal at the ResetError input.

Explanation of the parameters

Name:	**InputDelayTime**
Declaration:	INPUT
Data type	INT
Explanation	Delay time in milliseconds for commands and setpoints entered by hardware. Range of values: 0 or 1 ... 32000 [ms] A delay time of at least 1000 ms is recommended. 0 can be entered if the parameter is not required. For more detailed information on this parameter, refer to the section Function.

Name:	**MaxInputTime**
Declaration:	INPUT
Data type	INT
Explanation	Monitoring time in seconds for commands and setpoints entered by hardware. Range of values: 0 or 1 ... 32000 [s] A monitoring time of at least 30 s is recommended. 0 can be entered if the parameter is not required. For more detailed information on this parameter, refer to the section Function.

Name:	**ResetError**		
Declaration:	INPUT		
Data type	BOOL		
Explanation	Input for resetting the GlobalCmdOutputError output.		
	Range of values:	Input	I 0.0 ... I n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
	If the parameter is not required, specify a memory bit or data bit that always has signal 0.		
	For more detailed information on this parameter, refer to the section Function.		

Name:	**InputOK**		
Declaration:	OUTPUT		
Data type	BOOL		
Explanation	Hardware command or setpoint entry has been executed correctly.		
	Range of values:	Output	Q 0.0 ... I n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
	If the parameter is not required, specify a memory bit or data bit that can be used as a scratchpad memory.		
	For more detailed information on this parameter, refer to the section Function.		

Name:	**InputError**		
Declaration:	OUTPUT		
Data type	BOOL		
Explanation	An error has occurred related to the hardware command or setpoint input.		
	Range of values:	Output	Q 0.0 ... I n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
	If the parameter is not required, specify a memory bit or data bit that can be used as a scratchpad memory.		
	For more detailed information on this parameter, refer to the section Function.		

Name:	**GlobalCmdOutputError**
Declaration:	OUTPUT

Data type	BOOL		
Explanation	Group message: A command output error has been detected in a station.		
	Range of values:	Output	Q 0.0 ... I n.7
		Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
		Data bit	DBm.DBX 0.0 ... n.7
	If the parameter is not required, specify a memory bit or data bit that can be used as a scratchpad memory.		
	For more detailed information on this parameter, refer to the section Function.		

2.6.6 FC PartnerStatus

Function

The FC PartnerStatus can show the current status ’disrupted’ or ’OK’ for a maximum of 8 SINAUT subscribers.

The FC can be called at any point in the cyclic SINAUT program (in OB1).

If you want to monitor the status of more than 8 subscribers, an appropriate number of PartnerStatus FCs must be included in the SINAUT program.

A SINAUT subscriber (partner) can be an ST7 CPU or an ST7cc to which a connection was configured, or a local TIM.

One bit per subscriber is reserved in the PartnerStatus output byte to indicate the status of the respective subscriber:

FALSE (or 0): Subscriber is disrupted (or the corresponding input parameter is not being used, i.e. configured as 0, or subscriber is unknown).

TRUE (or 1): Subscriber OK

Explanation of the parameters

Name:	**Partner1 ... Partner8**	
Declaration:	INPUT	
Data type	INT	
Explanation	SINAUT subscriber number of the subscriber to be monitored	
	Range of values:	0 or 1 ... 32000
	0 = Dummy value for unrequired parameter	
	1 ... 32000 = Number of the subscriber to be monitored	

If a set subscriber number not found in the administration (in DB-BasicData), an entry to this effect is made in the diagnostic buffer only during startup (event ID B101). The CPU does not change to STOP. The status of a correctly configured subscriber is indicated in the PartnerStatus output byte; unknown subscribers are not processed until the parameter error has been corrected. Their status bits are set to 0.

Name:	**PartnerStatus**		
Declaration:	OUTPUT		
Data type	BYTE		
Explanation	Output byte for indicating the status of a subscriber to be monitored.		
	Range of values:	Output bytes	QB0 ... QBn PQB0 ... PQBn
		Memory bytes	MB0 ... MBn LB0 ... LBn
		Data bit bytes	DBm.DBB0 ... n

The assignment of the status bits in the PartnerStatus output byte in relation to the parameters **Partner1 ... Partner8**:

	.7	.6	.5	.4	.3	.2	.1	.0
Partner X	8	7	6	5	4	3	2	1

Status:
0 = subscriber **Partner X** disrupted or parameters not set or unknown
1 = subscribe **Partner X** OK.

2.6.7 FC PartnerMonitor

Function

FC PartnerMonitor displays important status information about a SINAUT subscriber (see PartnerStatus parameter). The FC can also be used to trigger a general request to the subscriber and to establish and disconnect a permanent connection to the subscriber.

The FC can be called at any point in the cyclic SINAUT program (in OB1).

If you want to monitor and control more than one subscriber, include an appropriate number of PartnerMonitor FCs in the SINAUT program.

A SINAUT subscriber (partner) can only be an ST7 CPU or an ST7cc to which a connection was configured. TIMs cannot be monitored or controlled by FC PartnerMonitor.

Note

FC PartnerMonitor can also be used in a station. However, the control inputs for establishing and terminating a permanent connection can then no longer be used. This only works in the master station, i.e. when the local TIM is a master TIM.

Explanation of the parameters

Name:	**PartnerNo**
Declaration:	INPUT
Data type	INT
Explanation	SINAUT subscriber number of the subscriber to be monitored and controlled.
	Range of values: 1 ... 32000 [ms]
	If the set PartnerNo is not found in the administration (in DB-BasicData), an entry to this effect is made in the diagnostic buffer only during startup (event ID B101). The CPU does not change to STOP. The status of a correctly configured subscriber is indicated in the PartnerStatus output word and the control inputs are processed. Unknown subscribers are not processed until the parameter assignment error has been corrected. The PartnerStatus output word remains set to 0 during this time.

Name:	**MaxConnectTime**
Declaration:	INPUT
Data type	INT
Explanation	Maximum connection duration for a permanent connection.
	Range of values: 0 (= no limit) or 1 ... 480 [minutes]
	If the time specified here is greater than 0, it is activated at the start of a permanent connection (see PermanentCall_On parameter). If the time elapses before the permanent connection is reset, it is automatically disconnected. The time is retriggered as long as the signal 1 is present at the PermanentCall_On input.
	The time specified here applies to a permanent connection in a dial-up network as well as to a permanent connection (continuous polling) on a dedicated line.

Name:	**PartnerStatus**	
Declaration:	OUTPUT	
Data type	WORD	
Explanation	Output word to indicate the status of the subscriber to be monitored.	
Range of values:	Output words	QW0 ... QWn PQW0 ... PQWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data bit words	DBm.DBW0 ... n
	If you do not require the parameter, simply leave it open.	
	The meaning of the status bits in the PartnerStatus output word:	

	Bit no.															
PartnerStatus	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0

Bit	.0	Status of the subscriber:
	0	0 = Subscriber disrupted
	1	1 = Subscriber OK

Bit	.1	Status of the redundant connection:
	0	0 = Redundant connection is disrupted
	1	1 = All connections OK.

Bit	.3	.2	Status of the general request (GR):
	0	0	0 = GR complete without error
	0	1	1 = GR started
	1	0	2 = GR start received
	1	1	3 = GR finished with error (GR incomplete or cannot be executed, e.g. due to disrupted subscriber)

Bit	.6	.5	.4	Status of the dial-up connection:
	0	0	0	0 = No connection
	0	0	1	1 = Outgoing call activated
	0	1	0	2 = Incoming call established
	0	1	1	3 = Outgoing call established
	1	0	0	4 = Permanent connection registered
	1	0	1	5 = Permanent connection established
	1	1	0	6 = Permanent connection disconnected

Bit	.7	Status of the dial-up connection:
	0	0 = No dial-up connection check in background
	1	1 = Dial-up connection check in background is activated

Bit	.8	Status of continuous polling (on dedicated line):
	0	0 = No continuous polling
	1	1 = Continuous polling activated

Bit	.9	Status of the WAN connection resources:
*)	0	0 = Sufficient resources on partner
	1	1 = Insufficient resources on partner

Bit	.10	Time status:
	0	0 = Date/time not available / not OK on partner
	1	1 = Date/time OK on partner

Bit	.11	Time synchronization:
	0	0 = The partner CPU received a plausible time during the last synchronization or no synchronization time has been received since startup.
	1	1 = The partner CPU has received an implausible synchronization time; the last valid time will continue to be used.

Note

Bit 11 cannot be displayed in conjunction with TimeTask version ≤ 1.6.

Note

The following parameters, GeneralRequest, PermanentCall_On and PermanentCall_Off are in/out parameters (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

Name:	**GeneralRequest**
Declaration:	IN_OUT
Data type	BOOL
Explanation	Input for triggering a general request to the subscriber specified with PartnerNo.

Range of values:	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

A general request to the subscriber is triggered with a 1 signal at this input if no request is active for this subscriber at this time. The input is then automatically reset by the FC. If an input of a digital input is specified (I 0.0 ... I n.7), the user is responsible for resetting the signal at the input. This must be done before ending the currently running general request otherwise another general request is triggered immediately.

Name:	**PermanentCall_On**
Declaration:	IN_OUT
Data type	BOOL
Explanation	Input for triggering a permanent connection to the subscriber specified with PartnerNo.

Range of values:	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

A permanent connection to the subscriber is triggered with a 1 signal at this input if there is currently no permanent connection to this subscriber. The input is then automatically reset by the FC. If an input of a digital input is specified (I 0.0 ... I n.7), the user is responsible for resetting the signal at the input. This should be done at the latest before terminating the existing permanent connection.

A 1 signal at the input PermanentCall_On also activates the time specified with MaxConnectTime if it is greater than 0.

Depending on whether the subscriber can be reached over a dial-up connection or a dedicated line, the command to establish the permanent connection is processed as follows and indicated at the PartnerStatus output:

For a dial-up connection:

A dial-up connection is established by the master TIM to the appropriate subscriber and, regardless of the data traffic, maintained until the terminate command is sent.
The current status of the permanent connection is indicated in the PartnerStatus output word with the bits 4 ... 6 (see PartnerStatus parameter).

For a dedicated line:

In this case the master TIM operates in polling mode with the stations. A permanent connection is implemented in this case by ’continuous polling’ of the subscriber. This is actually an intermittent poll to the subscriber; in other words, the other subscribers on the dedicated line network are still polled but the preferred subscriber is polled again after every poll to a ’normal’ subscriber .
The current status of the continuous polling is indicated by bit 8 in the PartnerStatus output word (see PartnerStatus parameter).

Note

A permanent connection cannot be established from a station. This control input cannot therefore be used when FC PartnerMonitor is used in a station.

Name:	**PermanentCall_Off**
Declaration:	IN_OUT
Data type	BOOL
Explanation	Input for triggering termination of an existing permanent connection to the subscriber specified with PartnerNo.

Range of values:	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

A permanent connection to the subscriber is terminated with a 1 signal at this input if there is currently a permanent connection to this subscriber. The input is then automatically reset by the FC. If an input of a digital input is specified (I 0.0 ... I n.7), the user is responsible for resetting the signal at the input. This should be done at the latest before establishing the permanent connection again.

Depending on whether the subscriber can be reached over a dial-up connection or a dedicated line, the command to terminate the permanent connection is processed as follows and indicated at the PartnerStatus output:

For a dial-up connection:

The existing dial-up connection is terminated by the master TIM but only after any pending data has been sent.
The current status of the permanent connection is indicated in the PartnerStatus output word with the bits 4 ... 6 (see PartnerStatus parameter).

For a dedicated line:

The master TIM deletes the registration for continuous polling of the corresponding subscriber. The polling cycle for all connected subscribers continues in normal mode.
The current status of the continuous polling is indicated by bit 8 in the PartnerStatus output word (see PartnerStatus parameter).

Note

Continuous polling can also be canceled on a dedicated line by instructing the master TIM to start continuous polling of another subscriber. The existing job is then replaced by the new one.

Note

A permanent connection cannot be terminated by a station. This control input cannot therefore be used when FC PartnerMonitor is used in a station.

2.6.8 FC ST7ObjectTest

Function

Calling FC ST7ObjectTest in the programming error OB121 prevents a CPU stop when the CPU receives a message with an unknown destination object number.

FC ST7ObjectTest checks why OB121 is called, i.e. what type of block is missing. If the missing block is a data block and this data block is an instance DB of a SINAUT object, then the CPU will not change to STOP.

The parameter StopInOtherCases allows the user to specify the reaction if a block other than a SINAUT instance DB is missing: Stop or continue operation.

For more detailed information on the programming error OB121 and background information on the use of FC ST7ObjectTest: Refer to the section 'SINAUT test routing in the programming error OB121'.

Explanation of the parameters

Name:	**StopInOtherCases**
Declaration:	INPUT
Data type	BOOL
Explanation	CPU will change to STOP if other errors occur
	Range of values: TRUE or FALSE
	This parameter allows you to specify what should occur in other error situations: Stop or continue operation when OB121 is called because another data block, an FB or an FC is missing.

2.6.9 FB SMS_Control

Function

The SMS typical FB SMS_Control is used to send event-triggered SMS messages (SMS: Short Message Service) to a mobile phone set in the parameter assignment.

The configured SMS texts as well as the telephone number of the mobile phone are contained in a DB SMS_Data that must exist on the CPU. This DB is generated using the SINAUT ST7 configuration tool (see chapter *Configuration software for SINAUT ST7*, section *Parameters for Single Subscribers*). The texts are assigned 1:1 to a contiguous bit array consisting of digital inputs, memory bits or data of a data block.

The SMS typical operates internally with a job list. This contains all of the SMS jobs that were acquired due to a change in the bit array. The SMS jobs remain in the job list until they are completed (with or without error). They are then deleted from the job list.

An SMS task is successfully completed when:

- An SMS has been successfully sent to the SMS control center (SMS-C) over a fixed network connection and this in turn has successfully forwarded the SMS to the mobile phone.
- An SMS not requiring acknowledgement has been sent to the SMS-C over a direct GSM connection.
- An SMS requiring acknowledgement has been sent to the SMS-C over a direct GSM connection and acknowledged by the mobile phone.

An SMS task is ended with an error when:

- An internal monitoring time has expired without the SMS being sent to the SMS-C.
- The validity period of the SMS job in the SMS-C and CPU has expired without the SMS having been sent to the mobile phone by the SMS-C within this period or, when acknowledgement is required, without the SMS being acknowledged within this period.

If an SMS is ended with an error, a diagnostic message is entered in the diagnostic buffer of the CPU.

The jobs remain in the job list of the SMS typical until they are completed correctly or ended with an error, or until the SMS typical is disabled (Enable = FALSE).

If the SMS typical is disabled while there are still SMS jobs in the job list, an entry is made in the diagnostics buffer of the CPU for every incomplete SMS job.. A delete job is transferred to the TIM for all SMS jobs that are still active.

Note

If a signal that triggers an SMS occurs more than once, the SMS is only repeated when the preceding SMS job has been completed.

Operation with two mobile phone numbers

The SMS typical offers the option of working with a main and a substitute mobile phone. If access to the SMS-C is over an M20 GSM module, the two mobile phones may also belong to different networks.

Once a selectable monitoring time has expired, all SMS messages that have been sent up until this time but not yet successfully completed are sent once again in the same order to the substitute mobile phone and the monitoring time is restarted.

If the SMS typical is still unable to complete the SMS jobs over the substitute mobile phone within the allotted period, it switches back to the main mobile phone and sends the incomplete SMS messages once again, etc.

Each SMS that is repeated is entered in the diagnostic buffer of the CPU at the time it is resent . As a counterpart to this, a success message is entered in the diagnostic buffer each time a repeated SMS is completed successfully.

The switchover between the main and backup mobile phone is not made for each individual SMS; this always involves all SMS messages of a DB SMSData being processed.

When there are no more SMS messages for processing in the SMS typical, the system automatically switches back to the main mobile phone number so that when the next event occurs that triggers an SMS, this is sent to the main mobile phone.

Note

There is no switchover to a disrupted SMS-C. In this case, the SMS typical acts as if it has been configured for only one mobile phone number. Only one of the two mobile phone numbers is active at any given time; the other one remains passive.

Effect of the validity period

The validity period for SMS tasks can be set using the SINAUT ST7 configuration tools (see Subscriber Administration, *SMS Configuration* tab, *SMS DB Data* dialog, *Valid period* parameter).

The time for the validity period is started as soon as the SMS can be delivered to the SMS-C. Within the validity period the SMS-C continues to attempt to deliver any SMS sent to it to the mobile phone. The retry intervals, however, may be very irregular.

When the validity period expires, the SMS messages in the SMS job list of the SMS typical are deleted and a diagnostic message is entered in the diagnostic buffer of the CPU. The SMS messages in the SMS-C are normally also deleted at this time. How exactly the validity period in the SMS-C is adhered to varies considerably from provider to provider.

Note

If the SMS-C can only be reached over a fixed network connection, the validity period in the SMS-C is always approximately 48 hours. The "validity period" parameter has no effect in this case. The configured validity period is then only applicable for the CPU and can be set to 11 hours and 55 minutes at the maximum.

Retry mechanisms

If the SMS job is passed from the CPU to the TIM, the TIM is responsible for the send retry when the SMS-C cannot be reached.

If the TIM is able to deliver the SMS message to the SMS-C, the SMS-C takes over responsibility for automatic retries when the mobile phone cannot be reached immediately.

If the SMS message has been sent to the SMS-C but this is unable to deliver it to the mobile phone, the typical (on expiry of the validity period) switches over to the backup mobile phone (if this has been set).

The same retry mechanism is used for an SMS message requiring acknowledgement when this SMS can be delivered to the mobile phone but no acknowledgement is returned within the current validity period. Once again, the SMS typical also switches to the backup mobile phone if it is available.

When a SMS cannot be delivered to the backup mobile phone, or, in the case of messages requiring acknowledgement, no acknowledgement is received, the system switches back and forth between the main and backup mobile phones. This continues until delivery is completed or an acknowledgement has been received, or until the SMS typical is disabled (ENABLE = FALSE)

Other functions

In addition to sending and managing SMS messages, the SMS typical also processes two globally valid functions. They are global because their execution is valid for all SMS jobs connected with DB SMS_Data that is processed by the SMS typical. The two global functions are:

- Status request for all SMS jobs of DB SMS_Data being processed by the typical.
  The typical sends the status request to the TIM responsible for the SMS transmission. This reports the current status of each SMS job in its memory back to the typical (status messages 2, 3 or 8). The TIM closes the sequence of status messages with status message 11 which indicates that no more status information is available.
  This status request function executes only during the startup of the CPU and then only when the SMS typical is enabled at this time (ENABLE = TRUE).
- Deletion of all SMS tasks of DB SMS_Data being processed by the typical.
  Deletion takes place at several places:
  - On the CPU: All jobs stored in DB SMS_Data.
  - On the TIM: All jobs stored in the TIM for DB SMS_Data.
  - In SMS-C: All jobs stored in SMS-C for DB SMS_Data (only when the SMS-C can be reached over a fixed network).
- The delete function is executed automatically by the typical in the following situations:
  - When the instance DB belonging to the typical is initialized.
  - When DB SMS_Data to be processed by the typical is initialized.
  - When a status change is detected at the ENABLE input of the typical.

The status request and the deletion function are time monitored. The duration of the monitoring time is determined by the SupervisionTime (900 sec. by default) of the corresponding subscriber object in DB BasicData.

If the supervision time expires before the status request or the deletion function can be completed correctly, a message is entered in the diagnostic buffer of the CPU and a group message "Supervision time has expired" is set at the Status output parameter of the SMS typical.

As long as the status request or the delete function is active, processing of the SMS messages is postponed and it is not possible to trigger a new SMS.

The current status of the status request or delete function is indicated at the Status output word of the SMS typical.

Explanation of the parameters

Name:	**MessageObjectDB**
Declaration:	INPUT
Data type	BLOCK_DB
Default	-

Explanation Monitoring time in seconds for commands and setpoints entered by hardware.

Here, you specify the DB SMS_Data to be processed by the typical.

Range of values: DB1 ... DBn (n depending on CPU type)

DB SMS_Data contains all data and SMS texts to be processed in a group of message or alarm bits. This data block can be conveniently configured by the user with the SINAUT ST7 configuration tool and filled out with the required data and texts

Name: **Enabled**
Declaration: INPUT
Data type BOOL
Default TRUE
Explanation Enable block processing.

Range of values:		
	TRUE or FALSE	
	No parameter specified: Default TRUE is valid	
	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7 L 0.0 ... L n.7
	Data bit	DBm.DBX 0.0 ... n.7

The user can enable and disable SMS processing with this input. For example, this input can be used to schedule timed enabling and disabling by a user program if the SMS messages should only be sent to standby personnel evenings, nights and/or at weekends.

When processing is enabled, the SMS send functions of the typical are activated. If processing is disabled after being enabled, all undelivered, unacknowledged SMS jobs are then deleted. The typical enters the actual disabled status only after this has taken place.

If you do not require the parameter, simply leave it open.

Name: **Status**
Declaration: OUTPUT
Data type WORD
Default 0
Explanation The SMS typical can be monitored using the status bits in this word.

Range of values:		
	Output words	QW0 ... Qwn PQW0 ... PQWn
	Memory words	MW0 ... MWn LW0 ... LWn
	Data words	DBm.DBW0 ... n

If you do not require the parameter, simply leave it open.

The meaning of the status bits in the Status output word:

	Bit no.															
Status	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0

Bit.0 There are two mobile phones
Bit.1 At least one SMS being processed
Bit.2 At least one SMS not yet acknowledged
Bit.3 At least one SMS being repeated
Bit.4 DB-specific delete job active
Bit.5 DB-specific status request job active
Bit.6 Backup mobile telephone activated
Bit.7 Monitoring time expired Delete or status request job ended with error (details in the diagnostic buffer).
Bit.8 SMS processing activated
Bit.9 SMS processing temporarily paused (for details see operating mode, bits 12...15)
Bit.10 Not used
Bit.11 Not used

Bit	.15	.14	.13	.12	Operating mode: Status of current processing
	0	0	0	0	0 = disabled
	0	0	0	1	1 = enabled, processing ready/in progress
	0	0	1	0	2 = blocked: global status request in progress
	0	0	1	1	3 = blocked: global deletion of an SMS-C in progress
	0	1	0	0	4 = blocked, initialization of the instance DB + DB SMS_Data in progress
	0	1	0	1	5 = blocked: global deletion of all SMS-C in progress
	..	..	..	..	6-11 = not used
	1	1	0	0	12 = blocked, global SMS-C switchover in progress
	1	1	0	1	13 = blocked, current SMS-C disrupted or cannot be reached
	1	1	1	0	14 = blocked, keyword at end of DB-SMS_Data incorrect
	1	1	1	1	15 = not used

No SMS messages can be accepted for processing when:

- DB-specific delete job active
- A DB specific status request is in progress e.g. when the CPU is being restarted
- The instance DB or DB SMS_Data is being initialized
- There must be a reaction to changes at the Enable input
- A switchover between the main and backup mobile phone is in progress or, in general, the change of a mobile telephone number including a new SMS transfer.
- Access to the SMS-C or the SMS-C itself is disrupted
- The keyword in DB SMS_Data cannot be found

The typical is then temporarily blocked and sets bit .9.

Name:	**ReloadMobilPhoneNo**
Declaration:	IN_OUT
Data type	BOOL
Explanation	If the telephone number or one or both mobile phones needs to be changed online, activation of the new telephone numbers can be triggered over this input.

Range of values:	TRUE or FALSE No parameter specified: Default value FALSE is valid	
	Input	I 0.0 ... I n.7
	Bit memory	M 0.0 ... M n.7
	Data bit	DBm.DBX 0.0 ... n.7

After the user has changed the mobile phone numbers in DB SMS_Data, the input can be set and the new telephone number data activated online.

Otherwise the typical always uses the telephone numbers stored in DB SMS_Data when the typical detects a signal change from FALSE to TRUE at the ENABLE input.

If you do not require the parameter, simply leave it open.

Note

This is an in/out parameter (declaration IN_OUT). It is difficult to specify local bit memory with this parameter type and this should not be used.

2.6.10 DB SMS_Data

Function

DB SMS_Data contains all data and SMS texts needed by the SMS typical FB SMS_Control to send and manage SMS messages. This data block can be conveniently configured by the user with the SINAUT ST7 configuration tool and filled out with the required data and texts

Using the information contained in DB SMS_Data, the user can read the current status separately for each SMS message and determine when the most recent message was last transmitted as an incoming or outgoing message (when a time stamp is being used) and, if the messages require acknowledgment, when the most recent acknowledgement occurred.

Structure of DB SMS_Data

The DB consists of a body with globally required information and one or more sections called SMS objects that contain the data required for each SMS message. The data of an SMS object are defined in UDT125 ShortMessageObject that is contained in the TD7 library.

The global information in the body of the DB includes:

- The SINAUT subscriber number of the main SMS control center (main SMS-C) and possibly a backup SMS control center.
- The telephone number of the main mobile phone and, if available, the telephone number of the backup mobile phone.
- The start address of the contiguous bit array (digital inputs, memory bits or data bits). If a bit in this field changes, the corresponding SMS message is sent.

Structure of an SMS object

The structure of an SMS object is defined in UDT125 ShortMessageObject. A ShortMessageObject is included in DB SMS_Data for each bit in the defined bit array.

The information stored here includes the following:

- The text of the SMS message to be sent
- When the most recent SMS was sent with an incoming or outgoing message
- The current typical-internal processing status of the SMS message
- The current processing status of the SMS message on the TIM / SMS-C, including the time stamp of the most recent status message and ID number assigned by the TIM (TIM ID) that allows the diagnostic buffer entries to be referenced (on the TIM and CPU)
- Various entries required by the typical for organizing processing

Each SMS object consists of a maximum of 82 data word (with a maximum of 122 characters of text). The structure is as follows:

Table 2- 7 Structure of an SMS object in DB SMS_Data

Number of data words	Short name		Meaning
Max. 62	SMS text		SMS text string, max. 122 characters long
1	Year	Month	Acquisition time stamp for the most recently sent incoming SMS
1	Day	Hours	
1	Minute	Second	
1	Year	Month	Acquisition time stamp for the most recently sent outgoing SMS
1	Day	Hours	
1	Minute	Second	
1	CurrentMessageStatus		Typical-internal SMS processing status (see description of CurrentMessageStatus)
1	Status incoming SMS	Status outgoing SMS	Most recent SMS status received by the CPU from the TIM or SMS-C (see description of the Status incoming SMS)
1	TIM ID for incoming SMS		ID number for incoming SMS assigned by the TIM
1	TIM ID for outgoing SMS		ID number for outgoing SMS assigned by the TIM
1	Year	Month	Time stamp of the status message most recently transferred by the TIM to the CPU
1	Day	Hours	
1	Minute	Second	
1	LastControlMessage		Most recent control instruction from the CPU to the TIM (without time stamp)
1	RemainingValidPeriodP		Time remaining in validity period of the incoming SMS
1	RemainingValidPeriodN		Time remaining in validity period of the outgoing SMS
1	SupervisionTime		Time remaining in the supervision time until incoming or outgoing SMS must be transfer to the SMS-C at the latest
1	PreviousMessageAddress		Previous address (for chained job list)
1	NextMessageAddress		Next address (for chained job list)
1	MessageNumber		Message number of the SMS for identification and diagnostics

CurrentMessageStatus

The meaning of the status bits in CurrentMessageStatus:

	Bit no.															
Status	.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0

Bit	.1	.0	Status of the SMS after incoming signal edge:
	0	0	0 = Job complete without error
	0	1	1 = TIM confirms: (incoming) SMS received
	1	0	2 = SMS-C confirms: (incoming) SMS received
	1	1	3 = Job ended with error

Bit	.3	.2	Status of the SMS after outgoing signal edge (only relevant when SMS is sent on both an incoming and outgoing message):
	0	0	0 = Job complete without error
	0	1	1 = TIM confirms: (outgoing) SMS received
	1	0	2 = SMS-C confirms: (outgoing) SMS received
	1	1	3 = Job ended with error

Bit.4	Edge memory bit. SMS entered in the job list on signal edge change
Bit.5	Incoming SMS: Job is being repeated
Bit.6	Outgoing SMS: Job is being repeated
Bit.7	No signal edge change processing as long as the SMS job is in progress (an incoming or outgoing SMS message is not sent again when the previous, identical message has not yet been completed).
Bit.8	SMS is disabled
Bit.9	Mandatory acknowledgment of SMS
Bit.10	SMS waiting of acknowledgment
Bit.11	SMS is prefixed with + / - for incoming or outgoing
Bit.12	Not used
Bit.13	Not used
Bit.14	Reserved
Bit.15	Reserved

Status incoming / outgoing SMS

The valid SMS processing status on the TIM is reported by the TIM to the CPU. SMS typical enters the return messages belonging to an incoming SMS message in the "Status incoming SMS" byte and those belonging to an outgoing SMS message in the "Status outgoing SMS" byte.

Status 1 is an exception here. It is not reported by the TIM to the CPU but is created by the SMS typical itself and entered in the status byte for incoming or outgoing SMS.

The following classifications are described below:

Status
Status information: is sent by the TIM to the CPU when there is a change in the status or following a status request.

Pulse
"Pulse" information: is sent once to the CPU due to an event on the TIM; it cannot be requested.

Fixed network
Status information valid for the fixed network access to the SMS-C over the analog telephone network or ISDN network.

Mobile network
Status information valid for the mobile network access to the SMS-C via an M20 module.

The following status entries are possible:

1. Job monitoring time has expired, SMS could not be sent in time to the SMS-C.
   Status, fixed network, mobile network
2. SMS not yet sent to SMS-C.
   Status, fixed network
   Comes as a response to a status request; cannot be requested for a mobile network access.
3. SMS saved in SMS-C but not yet delivered to the mobile phone.
   Status, fixed network
   This status cannot be requested for a mobile network access.
4. SMS successfully delivered to the mobile phone.
   Pulse, fixed network
   Counts as a ’completed without error’ message.
5. SMS cannot be delivered to the mobile phone.
   Pulse, fixed network
   Counts as a ’ended with error’ message.
6. SMS successfully delivered to the SMS-C.
   Pulse / status, mobile network
   Pulse for messages not requiring acknowledgement. Counts then as a ’completed without error’ message.
   Status for messages requiring acknowledgment. The TIM waits for an acknowledgment or a delete job. Does not count as a ’completed’ message.
7. SMS acknowledged by mobile phone.
   Pulse, mobile network
   Counts as a ’completed without error’ message.
8. SMS could not be sent in time to the SMS-C.
   Status, mobile network
   The TIM repeats the transmission until the SMS can be delivered or the SMS is deleted.
   Possible causes are:
   – "ERROR" message from M20 module
   – Monitoring time on the TIM has expired (the M20 module did not answer).
   – Disruption in the GSM network
9. SMS successfully deleted.
   Pulse, fixed network, mobile network
   Response of the TIM to a global delete job transferred by the CPU to the TIM. The TIM sends a separate status message 9 to the CPU for every SMS task deleted on the TIM.
   With a fixed network connection the SMS message is deleted in the SMS-C and in the administration of the TIM.
   With mobile network access it is not possible to delete the message in the SMS-C. It is only deleted in the TIM administration.
10. No entries to be deleted or all entries deleted.
    Pulse, fixed network, mobile network
    Response of the TIM to a global delete job transferred by the CPU to the TIM. In this case all entries for a DB SMS_Data
    are deleted or there are no entries for this DB SMS_Data.

11. No other entry is available or no entry exists at all.
    Pulse, fixed network, mobile network
    Response of the TIM to a global status request transferred by the CPU to the TIM. The TIM had previously transferred the current status of all remaining active entries to the CPU.
12. SMS could not be deleted in the SMS-C. Only the entry in the TIM records is deleted.
    Pulse, fixed network
    Response of the TIM to a specific delete job for an individual SMS. Delete job has was transferred from the CPU to the TIM.
    In this case a message in the SMS-C could not be deleted. The SMS in question probably no longer exists in the SMS-C or the SMS has already been delivered. Therefore only the corresponding entry in the TIM administration is deleted.

2.7 Test blocks

2.7.1 FC TestCopy

Function

Using FC TestCopy message traffic between SINAUT ST7 subscribers can be copied in part or whole. Individual message types can be specifically filtered using configurable search masks in the control field of the DB TestCopyData. These can then be copied from the send or receive buffers for further analysis in DB TestCopyData.

All send and receive messages are stored in the same DB TestCopyData. The chronological order of the copied send and receive messages is then easy to recognize.

The functions SendCopy (= log send messages) or RecvCopy (= log receive messages) can be individually activated or activated simultaneously but only one common operating mode (OM) is possible for both communication directions. The operating modes are set in data byte DBB0 of the DB TestCopyData, whereby the following assignments apply:

- DBB0, bits 0...3: Operating mode for RecvCopy function (operating mode 0, 1, 2 or 3)
- DBB0, bits 4..0.7: Operating mode for SendCopy function (operating mode 0, 1, 2 or 3)

If an operating mode > 0 is set in the less significant half-byte, this always applies to both communication directions. The value in DBX 4...7 (SendCopy function only) applies only if OM = 0 in bit 0...3 (no RecvCopy function required). Exception: DBB0 must be written with FF to delete DB TestCopyData, 0F is insufficient!

Examples:

DBB0 = 00h; no TestCopy-function activated

DBB0 = 03h; RecvCopy function only, mode 3, no SendCopy function required

DBB0 = 30h; SendCopy function only, mode 3, no RecvCopy function required

DBB0 = 33h; RecvCopy function and SendCopy function, mode 3, required

DBB0 = 23h; RecvCopy function and SendCopy function required, mode = 3

DBB0 = FFh; Delete content of DB TestCopyData

Prerequisites

The user program must meet the following conditions to use the TestCopy function:

- The FC TestCopy function must be present in the CPU;
- The DB TestCopyData in the CPU must be long enough. To ensure this, copy DB99 TestCopyData from the TD7 library to your CPU. Change the length of the buffer area as required by correspondingly increasing or decreasing the TestCopyBuffer area in the declaration header of the DB. This buffer area is preset as an array with a length of [0..240] WORDs.
- In the communication DB (type DB XComData, DB PComData or DB BComData) whose send and/or receive messages you wish to log, the following entries must be made (the best way is to use the predefined variable table VAT_TestCopy from the TD7 library):
  - The number of the DB TestCopyData must be entered in DW32 (symbolic name TestCopyDBNo).
  - The number of the FC TestCopy must be entered in DW34 (symbolic name TestCopyFCNo).

Including FC TestCopy in the user program

The test function is executed cyclically by FB XCom, FB PCom or FB BCom if the number of the FC TestCopy is entered in DBW34 of the data block DB XComData, DB PComData or DB BComData to be monitored.

Operating mode and filter settings for FC TestCopy

The user controls FC TestCopy using a VAT (VAriable Table, that is comparable to Force Variables in the S5 environment). This VAT has a predefined form and is available in the TD7 library as VAT_TestCopy.

The following settings in DB TestCopy are possible using VAT_TestCopy:

Name	Permitted values	Meaning
OperationMode	00 h	Function disabled.
	11 h	Message entry always at the beginning of DB TestCopyData.
	22 h	Write DB TestCopyData endlessly as a circulating buffer.
	33 h	Fill DB TestCopydata once, then set mode 0.
	FF h	Delete complete DB TestCopyData and preset defaults.
????_TgrmType	FF h	Copy all message types (MT = 0, 1, 2, 3) into DB TestCopyData.
	00 h	Copy only ORG messages (MT = 0).
	11 h	Copy only requested ORG messages (MT = 1).
	22 h	Copy only data messages (MT = 2).
	33 h	Copy only requested data messages (MT = 3).

Name	Permitted values	Meaning
	01 h	Copy messages with MT = 0 or MT = 1.
	23 h	Copy messages with MT = 2 or MT = 3.
	Any combination	Copy any combination of 0, 1, 2, 3.
????_DestSubscr	All perm. sbscr.	Filter the destination subscriber no. in message.
	-1	Copy all messages regardless of the destination subscriber no.
????_DestObject	All perm. obj.	Filter the destination object no. in message.
	-1	Copy all messages regardless of the destination object no.
????_SourceSubscr	All perm. sbscr.	Filter the source subscriber no. in message.
	-1	Copy all messages regardless of the source subscriber no.
????_SourceObject	All perm. obj.	Filter the source object no. in message.
	-1	Copy all messages regardless of the source object no.
????_StartIndex	All perm. ind.	Filter the start index no. in message.
	-1	Copy all messages regardless of the start index no.

FC TestCopy supplies a counted value in DBW12 of the DB TestCopyData indicating the number of message that has been received since the operating mode was set to 1, 2, or 3 and that matched the filter criteria; the number of sent messages is entered in DBW26.

In DBB28, the user receives a return value that provides information about the errors that occurred during processing of the FC. Up to now, the following have been defined:

RetVal = 0: No error

RetVal = 1: The specified DB TestCopyData is too short

RetVal = 10d: The operating mode entered in DBB0 is not defined

Notes on operation

The content of the DB TestCopyData is not deleted when the operating mode is changed; only internal pointers and message counters in the management area of DB TestCopyData are reset. It is therefore advisable to use the delete function "FF" when changing the operating mode to initialize the message buffer area with 0. This makes it easier to read the copied message blocks.

If you want to copy sent and received messages, the same operating mode must be entered in the left half-byte and the right half-byte of the OperationMode parameter. The following scheme applies:

Bit	.7	.6	.5	.4	.3	.2	.1	.0
	Mode for SendCopy				Mode for RecvCopy			

For modes 0, 1, 2 and 3:
If you want to delete the buffer, FFh should always be entered; separate deletion of the receive and send messages is not possible.

2.7.2 DB TestCopyData

Structure of DB TestCopyData

The following table shows the structure of the DB TestCopyData:

Address		Name	Format	Explanation
DBB	0	OperationMode	BYTE	Mode
DBB	1	Recv_TgrmType	BYTE	Receive filter: Message type (MT)
DBW	2	Recv_DestSubscr	INT	Receive filter: Destination subscriber no.
DBW	4	Recv_DestObject	INT	Receive filter: Destination object no.
DBW	6	Recv_SourceSubscriber	INT	Receive filter: Source subscriber no.
DBW	8	Recv_SourceObject	INT	Receive filter: Source object no.
DBW	10	Recv_StartIndex	INT	Receive filter: Start index no.
DBW	12	Recv_TgramCounter	INT	Number of copied received messages
DBB	14	SpareDBB14	BYTE	Reserve
DBB	15	Send_TgrmType	BYTE	Send filter: Message type (MT)
DBW	16	Send_DestSubscr	INT	Send filter: Destination subscriber no.
DBW	18	Send_DestObject	INT	Send filter: Destination object no.
DBW	20	Send_SourceSubscriber	INT	Send filter: Source subscriber no.
DBW	22	Send_SourceObject	INT	Send filter: Source object no.
DBW	24	Send_StartIndex	INT	Send filter: Start index no.
DBW	26	Send_TgramCounter	INT	Number of copied sent messages
DBB	28	RetVal	BYTE	Error information: 0 = No error 1 = DB TestCopyData too short 10 = Unknown operating mode
DBB	29	SpareDBB29	BYTE	Reserve
DBB	30	SpareDBB30	BYTE	Reserve
DBB	31	TestCopyStatus	BYTE	Status byte for Testcopy operation
DBB	32	TestCopyCmdByte	BYTE	Command byte for Testcopy operation
DBB	33	TestCopyDelCount	BYTE	Loop counter for TestCopy delete function
DBW	34	NextFreeCopyByte	INT	Address of the next free TestCopyBuffer byte
DBD	36	StartTimeSFC64	DINT	SFC64 time at the start of the copy procedure
DBB	40	TestCopyBuffer[0]	BYTE	Copy area, byte 0
DBB	41	TestCopyBuffer[1]	BYTE	Copy area, byte 1

Address		Name	Format	Explanation
DBB	42	TestCopyBuffer[2]	BYTE	Copy area, byte 2
DBB	43	TestCopyBuffer[3]	BYTE	Copy area, byte 3
:	:	:		:
DBB	xxxx	TestCopyBuffer[xxxx]	BYTE	Copy area, byte xxxx

The individual areas of DB TestCopyData

The DB for the TestCopy function is divided into several different areas:

- Area 1: DBB 0 ... DBB28
  User interface for setting the TestCopy operating mode and displaying any errors.
  This user interface in turn is divided into the following areas:
  - Area 1.1: DBB 1 ... DBB 13:
    Filter settings for RecvCopy function and number of counted received messages.
  - Area 1.2: DBB 15 ... DBB 27:
    Filter settings for SendCopy function and number of counted sent messages.
- Area 2: DBB 29 ... DBB 39:
  Internal management pointers
- Area 3: DBB 40 ... DBB xxxx:
  Buffer area for storing messages that match the filter criteria; the buffer area must be configured as an array [0...xxxx] of WORD.

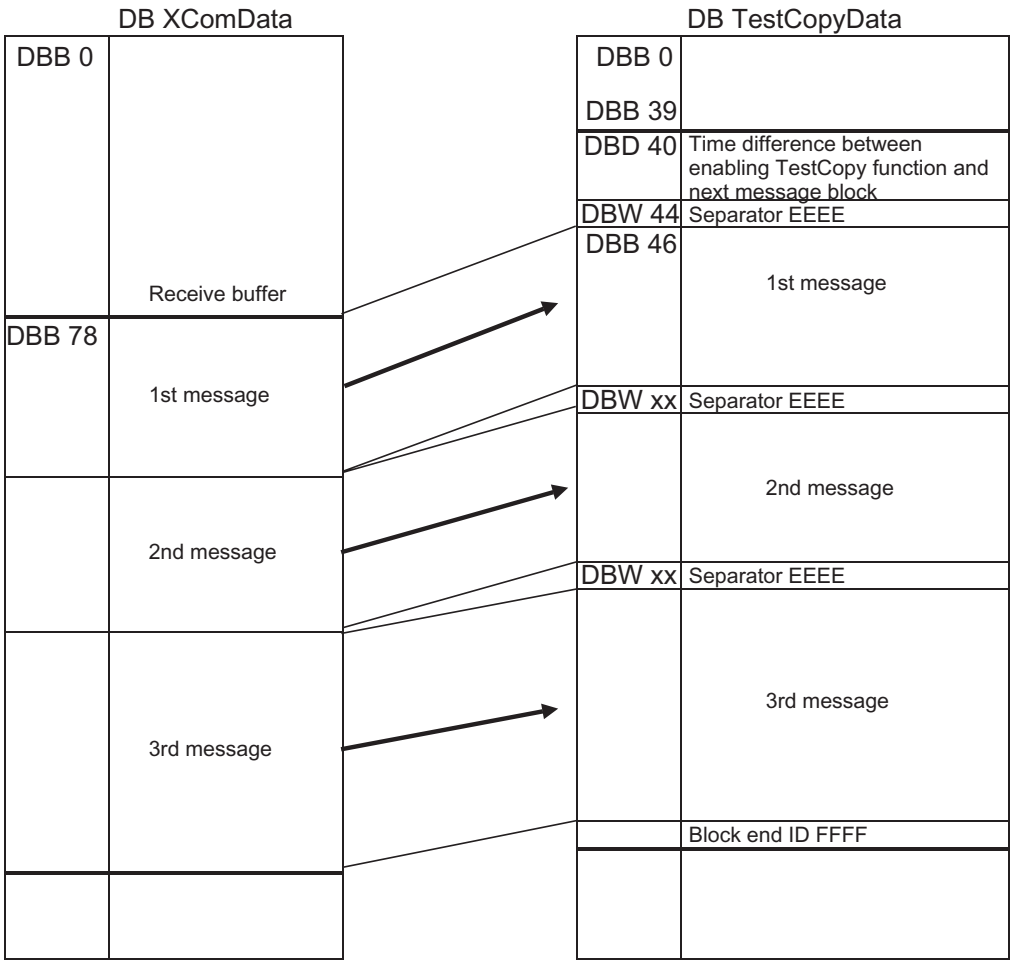
Structure of a copied message block

A message block can contain several messages. The messages are stored in DB TestCopyData according to the following rules:

1. The first entry indicates the time difference in milliseconds (7 decade BCD plus sign) since the last selection of an operating mode > 0.
2. This is followed by a separation sign AAAA for sent messages, EEEE for received messages.
3. Storage of the first message from the message block.
4. Separation sign AAAA, or EEEE:
5. Storage of the last message from the message block.
6. Block end ID FFFF.

Example

All received messages will be stored in DB TestCopyData. Communication is implemented using X blocks, i.e. a max. of 76 bytes per receive block. The receive buffer of the XComData DB is the source for FC TestCopy. The current receive block contains 3 messages.



Length calculation

FC TestCopy uses the following parameters for determining the minimum length for the DB TestCopyData:

Length of communication buffer	= LenComBuffer	= 76 or 202 bytes
Minimum message length	= LenMinTgrm	= 14 bytes
Offset management area	= Offset	= 40 bytes
Length of the time difference	= LenDt	= 4 bytes
Length of the block separators	= LenSeparator	= 2 bytes

The formula used for the actual calculation is the same for X communication and B communication. The results differ only due to different lengths for the communication buffer for X and B communication:

a) For X communication:

Len_{Min_Xcom} $=Len_{ComBuffer} + Offset + Len_{dt} + (Len_{ComBuffer} / Len_{MinTgrm} + 1) * Len_{Separator}$
$= 76 + 40 + 4 + (76 / 14 + 1) * 2$
$= 120 + 12 = 132$ bytes minimum

a) For B communication:

Len_{Min_Bcom} $=Len_{ComBuffer} + Offset + Len_{dt} + (Len_{ComBuffer} / Len_{MinTgrm} + 1) * Len_{Separator}$
$= 202 + 40 + 4 + (202 / 14 + 1) * 2$
$= 236 + 40 = 276$ bytes minimum

If FC TestCopy determines that the DB TestCopyData is not long enough, an error message to this effect is entered in data byte DBB28.

2.8 SFC / SFB system blocks used

Introduction

System functions, SFCs, and system function blocks, SFBs, are part of the operating system of the CPUs and are used by the TD7 blocks as auxiliary blocks. Since it should be possible to run the TD7 package on all CPU types, as a basic rule the only system blocks used are those that are available on all CPU types. Here, it is mainly the 300 CPUs that determine the system blocks that can be used. Presently there are only a few exceptions to this rule, namely, FC ListGenerator which has separate versions for S7-300 and S7-400 and FB BCom which is only used for S7-400 because communication function block connections are only possible there.

Because the system blocks are part of the operating system, they use no user memory.

Below, you will see an overview of the system blocks used that exist on all CPU types.

SFC 0 SET_CLK

Block for setting the date and time of day in the CPU.

SFC 1 READ_CLK

Block for reading the date and time of day on the CPU.

SFC 20 BLKMOV

Block for copying contiguous data areas.

SFC 22 CREATE_DB

Block for online generation of data blocks.

SFC 46 STP

Block for setting the CPU to 'STOP'.

SFC 52 WR_USRMSG

Block for entering a user message in the diagnostic buffer.

SFC 64 TIME_TCK

Block for reading the system time of the CPU.

SFC 65 X_SEND

Block for sending data over an unconfigured connection.

SFC 66 X_RCV

Block for receiving data over an unconfigured connection.

Other system blocks only available in 400 CPUs:

SFC 23 DEL_DB

Block for online deletion of data blocks.

SFC 24 TEST_DB

Block whose uses include determining whether a specific data block is available in the CPU.

SFC 25 COMPRESS

Block for online compression of the user memory, for example, after the deletion of a data block.

SFB 12 BSEND

Block for block-oriented sending of data over a configured connection.

SFB 13 BRCV

Block for block-oriented reception of data over a configured connection.

SINAUT Diagnostics and Service tool 3

3.1 Overview of the functions and operation of the SINAUT Diagnostics and Service tool

Introduction

The SINAUT Diagnostics and Service tool provides the user with functions for checking connections, interfaces and communication as well as the firmware and software components of the network subscribers of a SINAUT installation.
The most important functions are as follows:

- Reading the diagnostic data from a TIM or CPU module
- Reading the diagnostic buffer
- Checking and setting the module time
- Reading the module parameter assignment
- Activating a message trace
- Firmware update of TIM modules
- Downloading a new parameter assignment to the TIM modules

Note

Diagnostics functions that are also available in the SIMATIC Manager are described here with the emphasis on diagnostics of TIM modules.

3.1.1 Starting the program and types of access

Opening the Diagnostics and Service tool

The SINAUT Diagnostics and Service tool is opened in the Windows start menu *SIMATIC / SINAUT ST7 / Diagnostics and Service*.

You can access the module-specific diagnostic information alternatively over:

- *Accessible nodes*
- the *SINAUT subscriber list* of the STEP 7 project

NOTICE
Regardless of whether you access this information using *Accessible nodes* or the *SINAUT subscriber list*, unless you activate the PG routing function, you can only access subscribers of the subnet of the local MPI bus to which the PG is connected.

Access using *Accessible Nodes*

To access the diagnostic data using *Accessible Nodes*, follow the steps outlined below:

1. Click on the *Accessible Nodes* button
   or click on the *Project / Accessible Nodes* menu.
   The *Accessible Nodes* dialog opens.
2. In the *Selectable Nodes* dialog, select the required subscriber from the list of MPI addresses by clicking on it with the mouse.

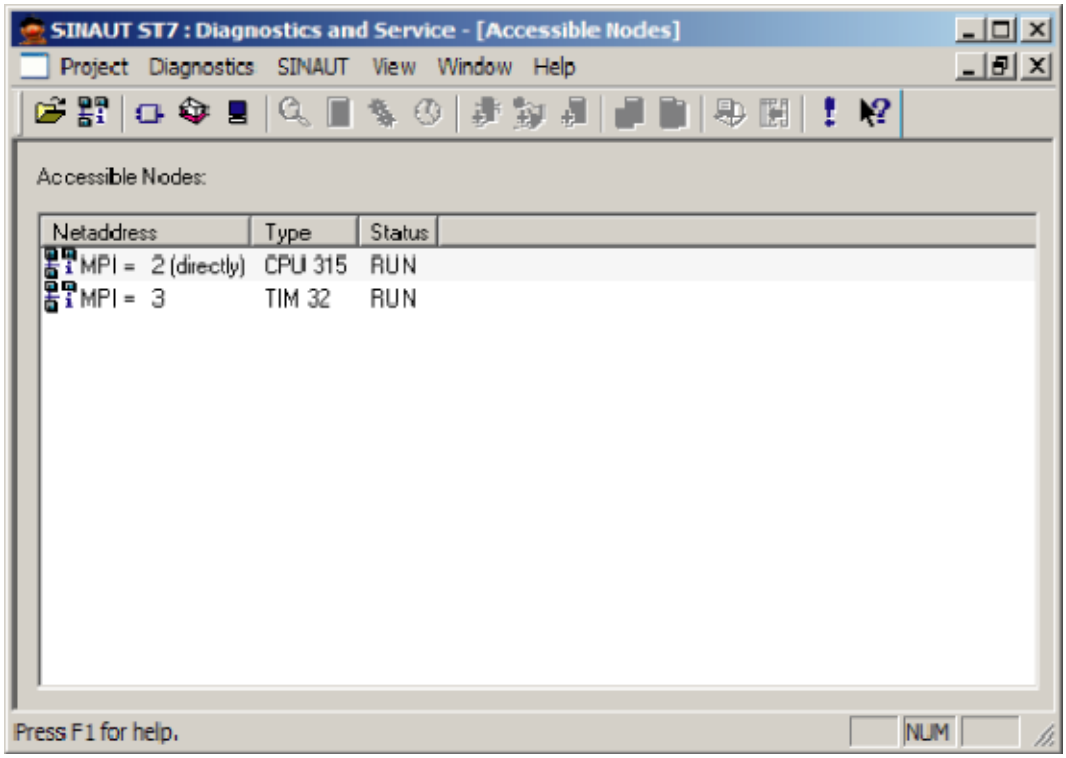


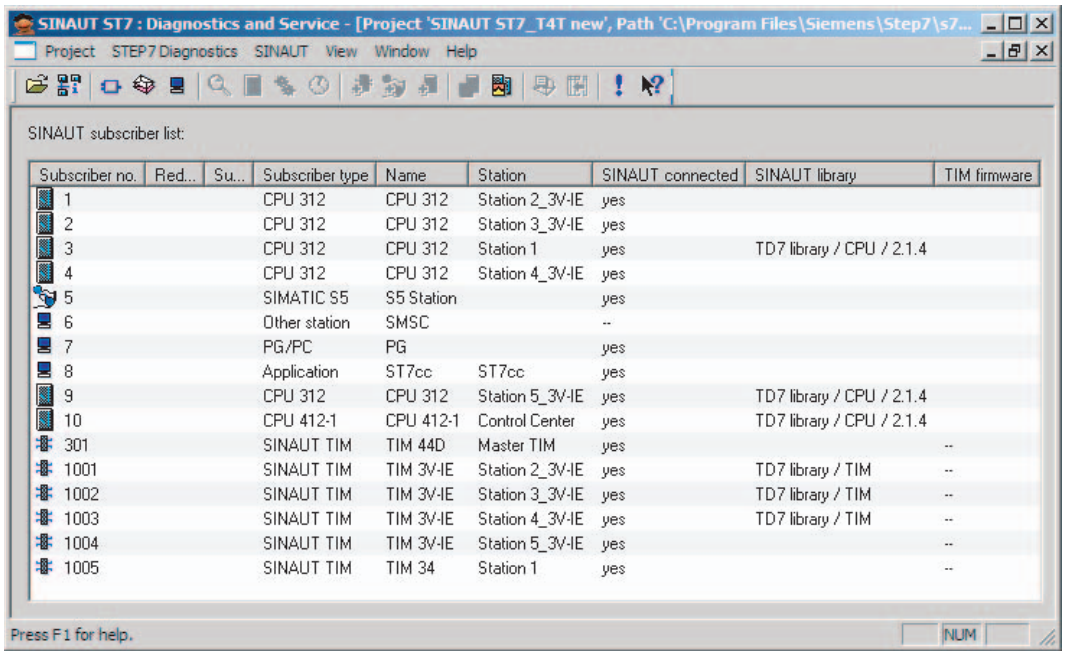
Figure 3-1 The *Accessible Nodes* dialog of a sample installation

Access using the *SINAUT subscriber list* of a STEP 7 project

To open the STEP 7 project in the Diagnostics and Service tool and to access the diagnostic data using the *SINAUT subscriber list*, follow the steps outlined below:

1. Click on the *Open Project* button in the toolbar or select the *Project / Open* menu. The *Open* dialog is displayed.
2. Select the STEP 7 project in the *User Projects* tab of the *Open* dialog and click on the *OK* button. The project window with the *SINAUT subscriber list* of the relevant project opens.

3. If the required project is not displayed in the *Open* dialog, click the *Browse* button. In the Browse dialog that opens, you can search for other projects and include them in the project list.
   As an alternative, you can open a current project with the *Project / Recently Used* menu.
4. Select the subscribers you require for the subsequent diagnostic functions in the *SINAUT subscriber list* by clicking on them with the mouse.


SINAUT ST7 : Diagnostics and Service - [Project 'SINAUT ST7_T4T new', Path 'C:\Program Files\Siemens\Step7\s7...
Project STEP7 Diagnostics SINAUT View Window Help
SINAUT subscriber list:
Subscriber no. Red... Su... Subscriber type Name Station SINAUT connected SINAUT library TIM firmware
1 CPU 312 CPU 312 Station 2_3V-IE yes
2 CPU 312 CPU 312 Station 3_3V-IE yes
3 CPU 312 CPU 312 Station 1 yes TD7 library / CPU / 2.1.4
4 CPU 312 CPU 312 Station 4_3V-IE yes
5 SIMATIC S5 S5 Station yes
6 Other station SMSC --
7 PG/PC PG yes
8 Application ST7cc ST7cc yes
9 CPU 312 CPU 312 Station 5_3V-IE yes TD7 library / CPU / 2.1.4
10 CPU 412-1 CPU 412-1 Control Center yes TD7 library / CPU / 2.1.4
301 SINAUT TIM TIM 44D Master TIM yes --
1001 SINAUT TIM TIM 3V-IE Station 2_3V-IE yes TD7 library / TIM --
1002 SINAUT TIM TIM 3V-IE Station 3_3V-IE yes TD7 library / TIM --
1003 SINAUT TIM TIM 3V-IE Station 4_3V-IE yes TD7 library / TIM --
1004 SINAUT TIM TIM 3V-IE Station 5_3V-IE yes --
1005 SINAUT TIM TIM 34 Station 1 yes --
Press F1 for help.
NUM
Figure 3-2 The *SINAUT subscriber list* of a sample project

Note

Attempting to access a remote subscriber using the *SINAUT subscriber list* of a STEP 7 project can lead to "misunderstandings" if the subscriber is not connected to the local MPI bus and the PG routing function is not activated. With functions involving access to the module, the remote subscriber is displayed in the *Path* field of the diagnostics dialog, however the diagnostic data is that of the locally connected subscriber.

The SINAUT subscriber list displays the following entries for each subscriber:

- **Subscriber no.**: The subscriber number of the SINAUT subscriber that is unique throughout the project
- **Red. Subscriber no.**: The *redundant subscriber number* parameter is used only when there is a redundant partner for the subscriber in question. The number specifies the common subscriber number under which the redundant system can be addressed by other subscribers.
- **Subscriber no. of red. Partner**: The *Subscriber number of the redundant partner* parameter is used only when there is a redundant partner for this subscriber. The parameter specifies which of the subscribers belong to a redundant relationship.
- **Subscriber type**: The *subscriber type* specifies the class of subscriber involved. The subscriber type cannot be set by the user.

- **Name**: The module, application or PC/PG name. This can be changed in the configuration. As default, this is the name of the module type or the application as specified in the configuration.
- **Station**: Name of the station specified by the user in the configuration using NetPro.
- **SINAUT connected**: Specifies whether a SINAUT connection was configured for the subscriber.
- **TD7 library version**: With CPU modules and modules of the type TIM 3V-IE, the name of the SINAUT system library for the TD7 software blocks is displayed.
- **TIM firmware version**: With TIM modules, the version of the TIM firmware is displayed.

PG Routing

If you connect a programming device (PG) or a PC to access the diagnostic data, you only have access to the local MPI network. The diagnostic data of remote subscribers in other subnets is not accessible.

To access subscribers in other subnets, you can use PG Routing. If you access data in a subordinate subnet after activating PG Routing, remember that you can only access subordinate subnets and not subnets higher in the network hierarchy. The prerequisites, functions and activation of PG routing are described in a separate chapter.

The PG Routing function is possible only when using the *SINAUT subscriber list*; PG Routing is not possible when using *Accessible Nodes*.

3.1.2 Access to SINAUT subscribers and working with the diagnostics dialogs

Activating diagnostic functions

The diagnostic functions are activated as follows:

1. Select a SINAUT subscriber by clicking on it with the mouse in *Accessible Nodes* or in the *SINAUT subscriber list* of a STEP 7 project.
2. Start the required diagnostic functions with one of the following alternatives:
   - Clicking on the corresponding button in the toolbar
   - Selecting the function in the *Project*, *STEP 7 Diagnostics* or *SINAUT* menus
   - Pressing the relevant function key
   - Right-clicking on the subscriber in *Access of Nodes* or in the *SINAUT subscriber list*. After clicking on the subscriber, select the required function with the right mouse button in the displayed context menu.
3. The dialog belonging to the selected diagnostic function is displayed.

Working with the dialogs

The graphic user interface of the SINAUT Diagnostics and Service tool is designed based on Windows technology. To use diagnostic functions, you must generally first select a particular subscriber or a component from a list in the Windows and dialogs and the function will then be executed and the diagnostic data displayed for this subscriber or component. The

function is then activated from a menu or by selecting a button and a dialog for the specific diagnostic function then opens.

When *selecting* a menu, a subscriber, or object is described, this involves clicking on the object once within the left mouse button.

Buttons found in many of the diagnostics dialogs are explained here and not in each subsection. These include the buttons:

- *Print:*
  Starts a printout of the currently open dialog.
- *Update:*
  Updates the content of the dialog with the current diagnostic data of the selected subscriber.
- *Save:*
  Saves the content of the open dialog in a file. You can select any directory and file name in the *Save* dialog.
- *Load:*
  Loads the diagnostic data relevant to the current dialog content from a previously saved file into the open dialog. The loaded diagnostic data is displayed in the dialog.

  Note

  When *loading* data from a file, the current project data in the dialog is overwritten by the data from the file. To display the data of the connected subscriber again, the dialog must be closed and reopened, in some cases, the display can be updated with the data of the connected subscriber again using the *Update* button.

- *Close:*
  Closes the current dialog. You return to the *Accessible Nodes* or *SINAUT subscriber list*.
- *Help:*
  Opens the online help function for the currently selected diagnostic function.
- *OK:*
  Confirms the entries made and closes the dialog.
- *Cancel:*
  Discards the entries made and closes the dialog.

3.1.3 Functions of the Diagnostics and Service tool

Overview of the diagnostic and service functions

The diagnostic functions of the SINAUT Diagnostics and Service tool can be grouped together as follows:

- STEP 7 diagnostics
- SINAUT diagnostics (TIM status information and TD7 software diagnostics)
- Message protocol diagnostics
- Service functions

The following table shows the diagnostic and service functions and all the menus in which the functions of the SINAUT Diagnostics and Service tool can be called.

The two right-hand columns in the table indicate that the scope of information when using the *SINAUT subscriber list* of a STEP 7 project is greater than when using *accessible nodes*.

Table 3- 1 Overview of the diagnostic and service functions of the SINAUT Diagnostics and Service tool

Function group, diagnostic function (remarks)	**Subscriber type relevant for diagnostics**	**Called in menu**	**Access over STEP 7 project**	**Access using Accessible Nodes**
STEP 7 diagnostics		**STEP 7 diagnostics**		
CPU messages	CPU, TIM	"	X	X
Module information (including messages in diagnostic buffer)	CPU, TIM	"	X	X
Operating mode	CPU, TIM	"	X	X
Setting the time	CPU, TIM	"	X	X
SINAUT				
SINAUT diagnostics		**SINAUT**		
TIM Diagnostics	TIM	"	X	X
TIM subscriber diagnostics	TIM	"	X	X
TIM diagnostic messages	TIM	"	X	
TIM Message Monitor	TIM	"	X	
TD7 software diagnostics		**SINAUT**		
TD7 CPU diagnostics (TD7 messages in diagnostics buffer)	CPU	"	X	X
TD7 block structure (configured data)	CPU	"	X	
TD7 block structure for all CPUs (configured data)	CPU	"	X	
TD7 CPU program comparison (configured data)	CPU	"	X	
TD7 communication configuration check (configured data)	CPU	"	X	
TD7onTIM diagnostics	TIM	"	X	
SDB Viewer	CPU, TIM	"	X	X
Service functions		**SINAUT**		
Download SDB	TIM	"	X	
Firmware update	TIM	"	X	
Repair	TIM	"	X	
Message protocol diagnostics		**Project**		
Testcopy DB	CPU	"	X	X
TIM message protocol	TIM	"	X	X
ST7cc / ST7sc message protocol	TIM	"	X	X

The diagnostic information is displayed only for SIMATIC CPU modules and SINAUT TIM modules.

3.2 STEP 7 diagnostics

Introduction

STEP 7 diagnostics involves the standard diagnostic functions of SIMATIC STEP 7. Over and above SIMATIC installations, the STEP 7 diagnostics in the SINAUT Diagnostics and Service tool provides information not only on the CPU modules but also information on the TIM modules of a project. The functions in the SINAUT Diagnostics and Service tool that are implemented on the TIM module:

- CPU messages
- Module information
- Operating mode
- Setting the time

3.2.1 CPU messages

Description of the functions

The *CPU messages* function is used to archive diagnostic messages entered by a CPU or TIM module in its diagnostic buffer. Without archiving, messages in the ring buffer of the CPU or TIM would be successively overwritten once the buffer is full.

The *CPU messages* function registers the PG used for diagnostics with one or more modules. The modules then transfer all newly generated diagnostic messages to the registered PG. The diagnostic messages of one or more modules are archive in a common list on the PG. The archive is designed as a ring buffer. The oldest messages are overwritten by newly arriving messages once the archive is full.

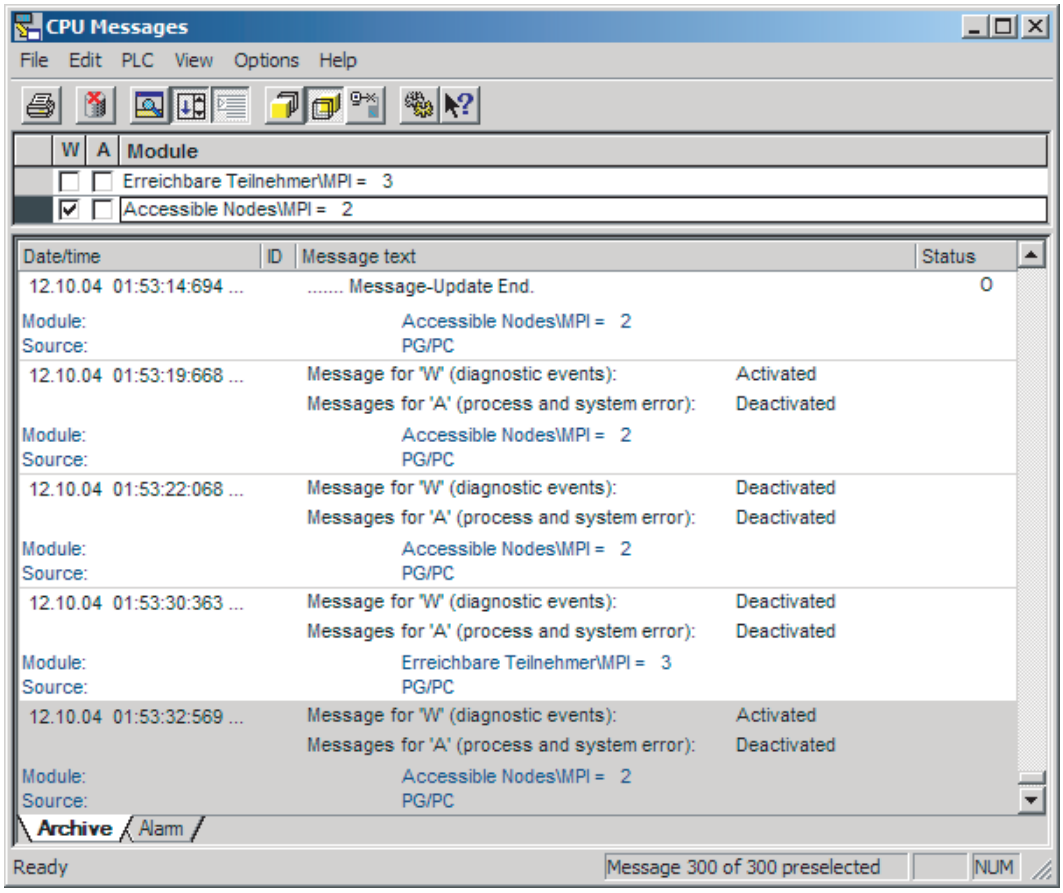

CPU Messages
File Edit PLC View Options Help
W A Module
Erreichbare Teilnehmer\MPI = 3
Accessible Nodes\MPI = 2
Date/time ID Message text Status
12.10.04 01:53:14:694 ... . Message-Update End. 0
Module: Accessible Nodes\MPI = 2
Source: PG/PC
12.10.04 01:53:19:668 ... Message for 'W' (diagnostic events): Activated
Messages for 'A' (process and system error): Deactivated
Module: Accessible Nodes\MPI = 2
Source: PG/PC
12.10.04 01:53:22:068 ... Message for 'W' (diagnostic events): Deactivated
Messages for 'A' (process and system error): Deactivated
Module: Accessible Nodes\MPI = 2
Source: PG/PC
12.10.04 01:53:30:363 ... Message for 'W' (diagnostic events): Deactivated
Messages for 'A' (process and system error): Deactivated
Module: Erreichbare Teilnehmer\MPI = 3
Source: PG/PC
12.10.04 01:53:32:569 ... Message for 'W' (diagnostic events): Activated
Messages for 'A' (process and system error): Deactivated
Module: Accessible Nodes\MPI = 2
Source: PG/PC
Archive Alarm
Ready
Message 300 of 300 preselected
NUM

Figure 3-3 *CPU Messages* dialog

The messages for diagnostic events are entered at the bottom of the dialog in the *Archive* tab of the message list.

From the menu of the dialog or using the buttons of the toolbar, various user-specific settings can be made for message output such as emptying the archive, processing messages, the view of the message window, the settings for the archive size and saving the PG connections to the registered modules for the next time the *CP messages* function is called.

Operator activities

1. Select a subscriber by clicking on it in the *SINAUT subscriber list* of the open project or in *Accessible Nodes*.
2. Open the *CPU Messages* dialog by selecting the *STEP 7 Diagnostics / CPU Messages* menu.
3. To register the PG/PC for the *CPU Messages* function, select the module in the *W column* of the module list at the top of the dialog. After the registration, the connection option (check box) of the module is selected in the *W column*. All the generated diagnostic messages of the module are then displayed in chronological order in the *Archive* tab of the message list at the bottom of the dialog.
   If no connection can be established to the subscriber, a symbol is displayed in the first column of the module list indicating that the connection is interrupted.

4. Click on the relevant field for the module in the *W column* of the module list again to deactivate archiving of the diagnostic messages.
5. Select the menu or the button of the dialog to change the settings.
6. Close the *CPU Messages* dialog by clicking on the *close dialog* button (x) in the title bar or double-clicking on the dialog name in the title bar of the dialog.

Closing the dialog deactivates the *CPU Messages* function.

3.2.2 Module information

Description of the functions

The *module information* function reads diagnostic data from the module of the connected station. The diagnostic data is displayed for the specific module in a series of tabs:

- *General* tab
  List of hardware and firmware components with their versions and information on the status of the CPU module
- *Diagnostic Buffer* tab
  List of diagnostic messages
- *Memory* tab
  Information on the utilization of the load and work memory.
- *Time System* tab
  Information on the data, time, time system and time synchronization as well as on the operating hours counter of CPU modules
- *Performance Data* tab
  Lists of the organization blocks, system blocks and address ranges
- *Communication* tab
  Information on transmission speeds, connection resources and cycle load caused by a communication

Further tabs are displayed for CPU modules:

- *Cycle Time* tab
  Set and measured cycle times of CPU modules
- *Stacks* tab
  Information on the content of the blocks stack (B stack), interrupt stack (I stack) and local data stack (L stack) of CPU modules

Operator activities

1. Select a local subscriber in the *SINAUT subscriber list* of the open project or in *Accessible Nodes*.
2. Open the dialog by selecting the *STEP 7 Diagnostics / Module Information* menu.
3. Select the individual tabs with the mouse.

General tab

The *General* tab displays the operating mode of the local CPU module and the operating mode and status of the connected module if this is selected for outputs of diagnostic data. The *Status* text box displays information on the status of the connected module from the perspective of the local CPU module. The following possible statuses are distinguished:

- Status *OK*: Module exists, access possible
- Status *Error*: Problem, access to module not possible (parameter assignment or access error)

Information on the module name and system identification is also display and in the *Version* output box, you will see a list of hardware and firmware components of the module with their order numbers or the name and version.

This is followed by information on the rack, address and slot of the CPU module.

Diagnostic Buffer tab

The *Diagnostic Buffer* tab displays the content of the diagnostic buffer of the module with information on the message number, time of day, date and event. The entries are sorted in descending chronological order; in other words, the latest message is at the top.

For the TIM, the last 50 entries of the diagnostic buffer are displayed, for a CPU normally the last ten diagnostic messages.

For TIMs, or diagnostic messages are displayed in plain language.

For CPUs, the system diagnostic messages are displayed as plain language and the TD7 diagnostic messages (in other words the messages created by the SINAUT user program) are displayed in hexadecimal format.

The station number (STA no.) listed with some messages in the *Details on Event* text box is the WAN network address of the relevant SINAUT network.

Note

If you have selected a CPU and want to see the plain text equivalent of TD7 diagnostic messages displayed in hexadecimal format in the *Diagnostic Buffer* tab, select the *TD7 CPU Diagnostics* function for the same CPU.

You will then see the same TD7 diagnostic messages

- in the *Module Information / Diagnostic Buffer* dialog in hexadecimal format and
- in the *TD7 CPU Diagnostics* dialog as plain text.

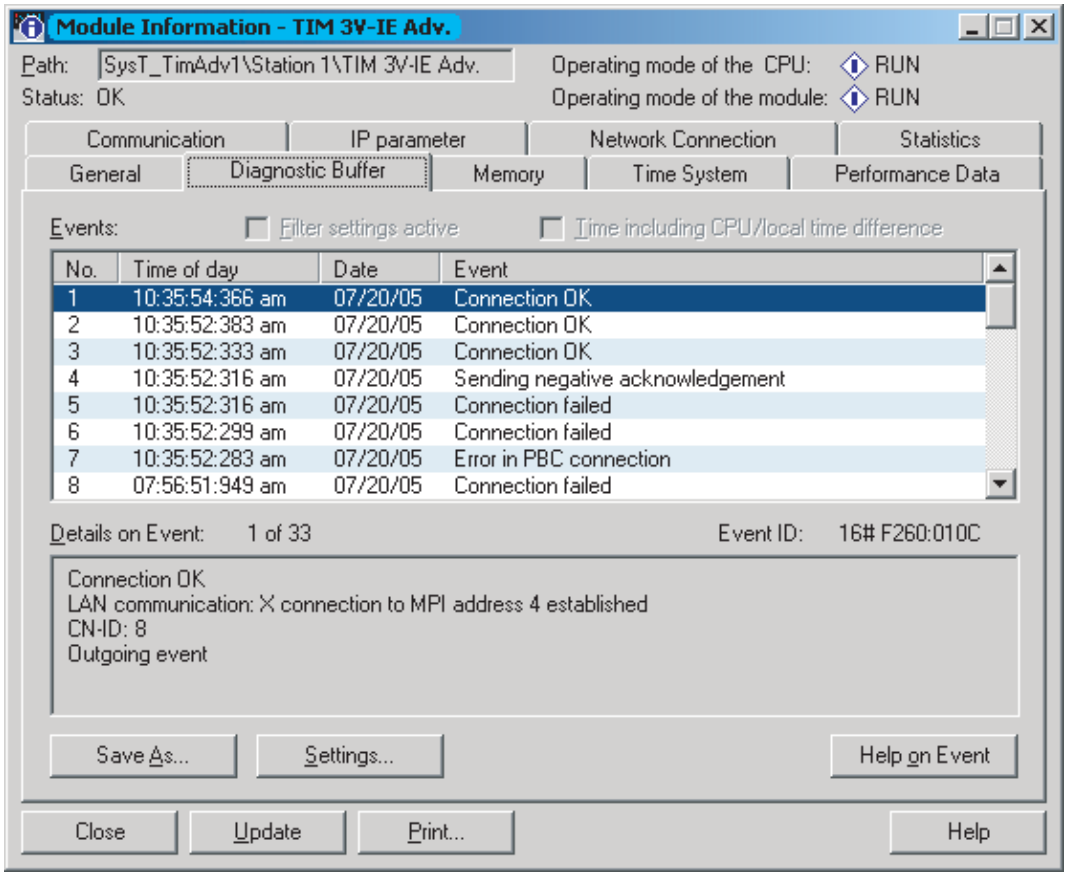


Figure 3-4 *Module Information* dialog, *Diagnostic Buffer* tab

To change the settings and select the event types of the message display in the *Diagnostic Buffer* tab, follow the steps outlined below:

1. Select the *Settings* button to open the *Setting for Display Diagnostic Buffer* dialog. The default number of entries can be changed either for CPUs or TIMs.
2. In the *Display Events* box, select or deselect the event types for message output. The selection is displayed or hidden.
3. In the lower part of the dialog, select the following options if necessary:
   - *Output event information in hexadecimal format*
   - *Update display during operating mode transition*
   - *Save settings for this dialog box*
4. Confirm your settings by clicking on the *OK* button or to discard the settings, click on *Cancel*. You then return to the *Diagnostic Buffer* tab.

Memory tab

The *Memory* tab displays information on the utilization of the free and assigned load memory and work memory.
The work memory utilization of a TIM of approximately 90% is normal and adequate for the TIM to function.

Time System tab

The *Time System* tab provides information on the time system of the module in three boxes:

- The current state and time of the module, its resolution and the existence of a real-time clock
- Time-of-day synchronization (CPU only)
- Run-time meter (CPU only)

Performance Data tab

The *Performance Data* tab does not contain any diagnostic information relevant to TIM modules. For CPU modules, information is displayed on organization blocks (OB), system blocks (SFC, SFB) and address ranges.

Communication tab

The *Communication* tab displays the following information:

- Maximum and unused connection resources for
  - PG communication
  - OP communication
  - S7 basic communication
- Configured cycle load due to communication. For a TIM, this is 100%.

Information on a communication relates only to the CPU.

IP Parameter tab

The *IP Parameters* tab displays the most important IP parameters of an Ethernet TIM:

- IP address: Configured IP address of the module.
- Subnet mask: Configured subnet mask of the module.

- Default router: If a default router was specified during configuration, the IP address of the default router is displayed here.
- IP settings: Indicates where the module obtained the IP parameters from.

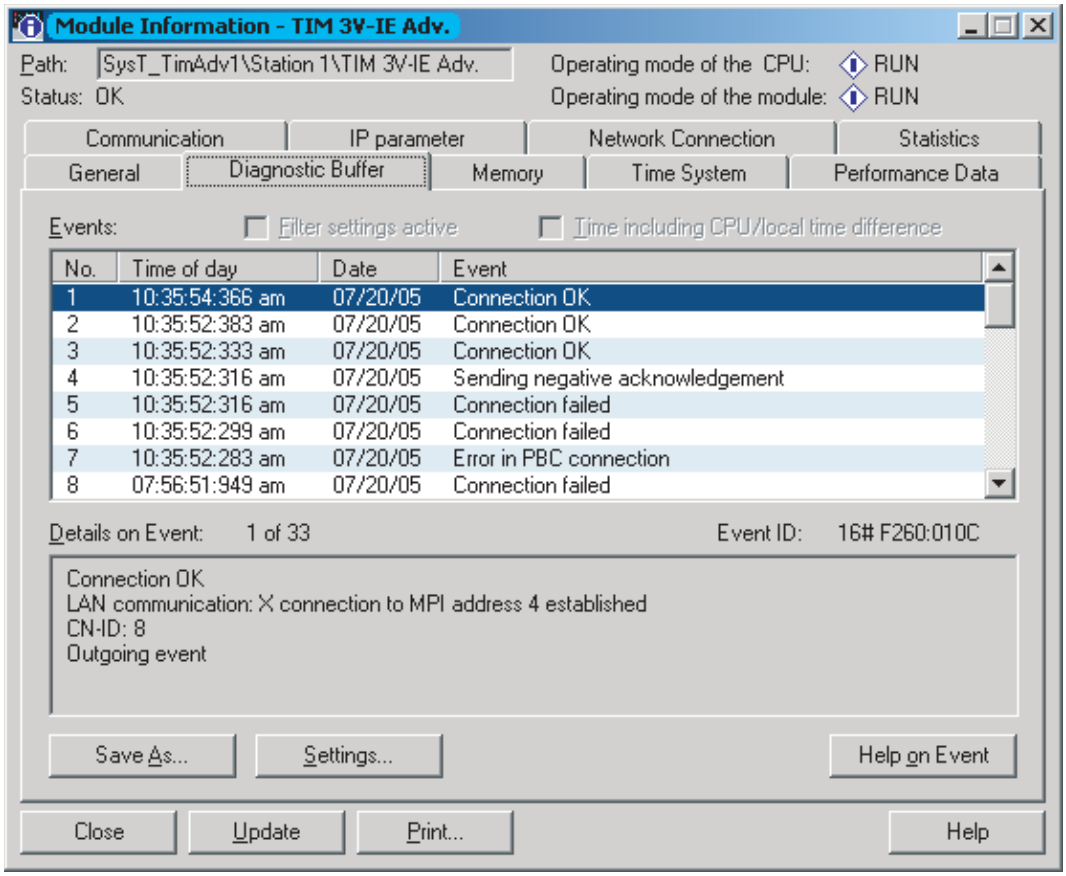


Figure 3-5 *Module Information* dialog, *IP Parameter* tab

Note

With a TIM 4R-IE, only information on the first Ethernet port P1 is displayed. For an overview of the status and parameters of both Ethernet ports of the module, refer to SINAUT Diagnostics, *IP Parameters* tab.

Network Connection tab

The *Network Connection* tab for an Ethernet TIM displays the MAC address of the module and information on the status and settings of the Ethernet port:

- Link Status: Indicates whether or not a physical connection to Ethernet exists.
- Settings: Shows the setting for detecting network settings, here: "Automatic" (Autosensing)
- Mode: Indicates the transmission speed and duplexity on Ethernet.

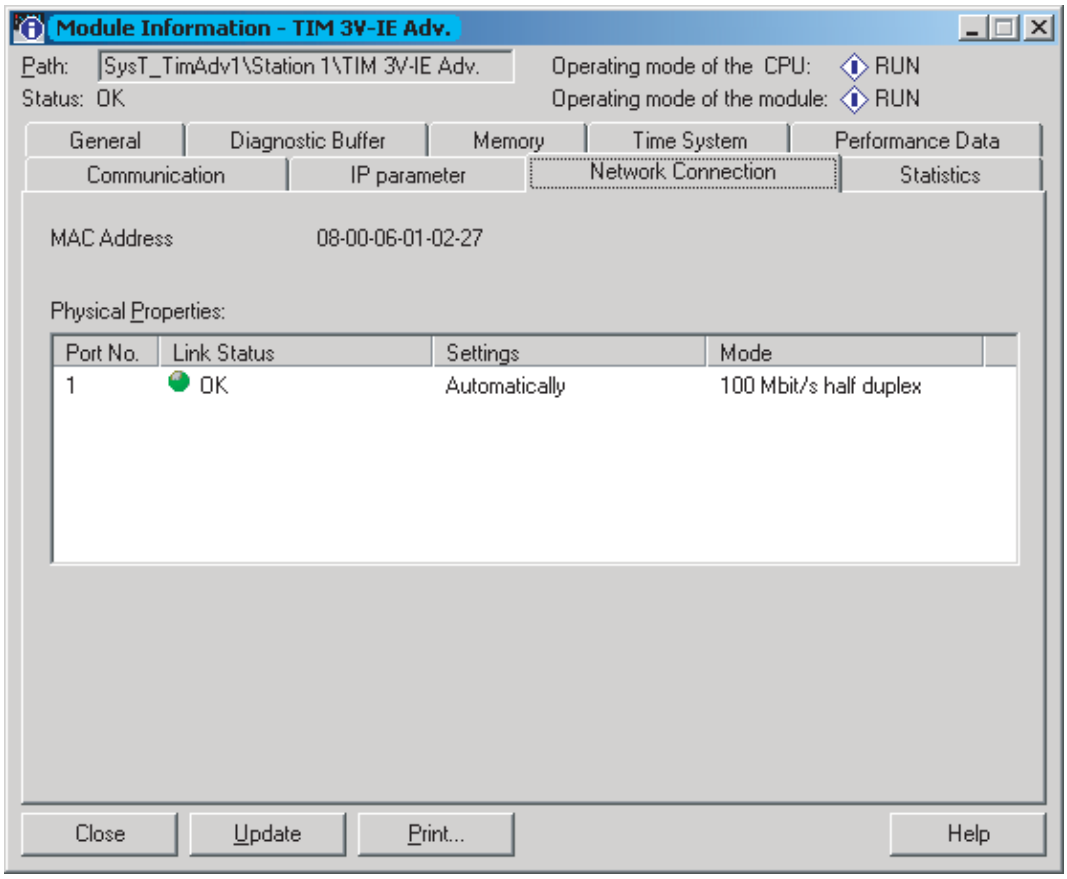


Figure 3-6 *Module Information* dialog, *Network Connection* tab

Note

With a TIM 4R-IE, only information on the first Ethernet port P1 is displayed. For an overview of the status and parameters of both Ethernet ports of the module, refer to SINAUT Diagnostics, *IP Parameters* tab.

Statistics tab

This tab is available only for Ethernet TIMs.

The *Statistics* tab contains transmission statistics for the Ethernet ports. The number of transferred data packets with and without errors since the last reset or restart of the module

is displayed for the send and receive directions. This time time is displayed as module time in the tab. The statistical values can be reset to zero with the *Reset* button.

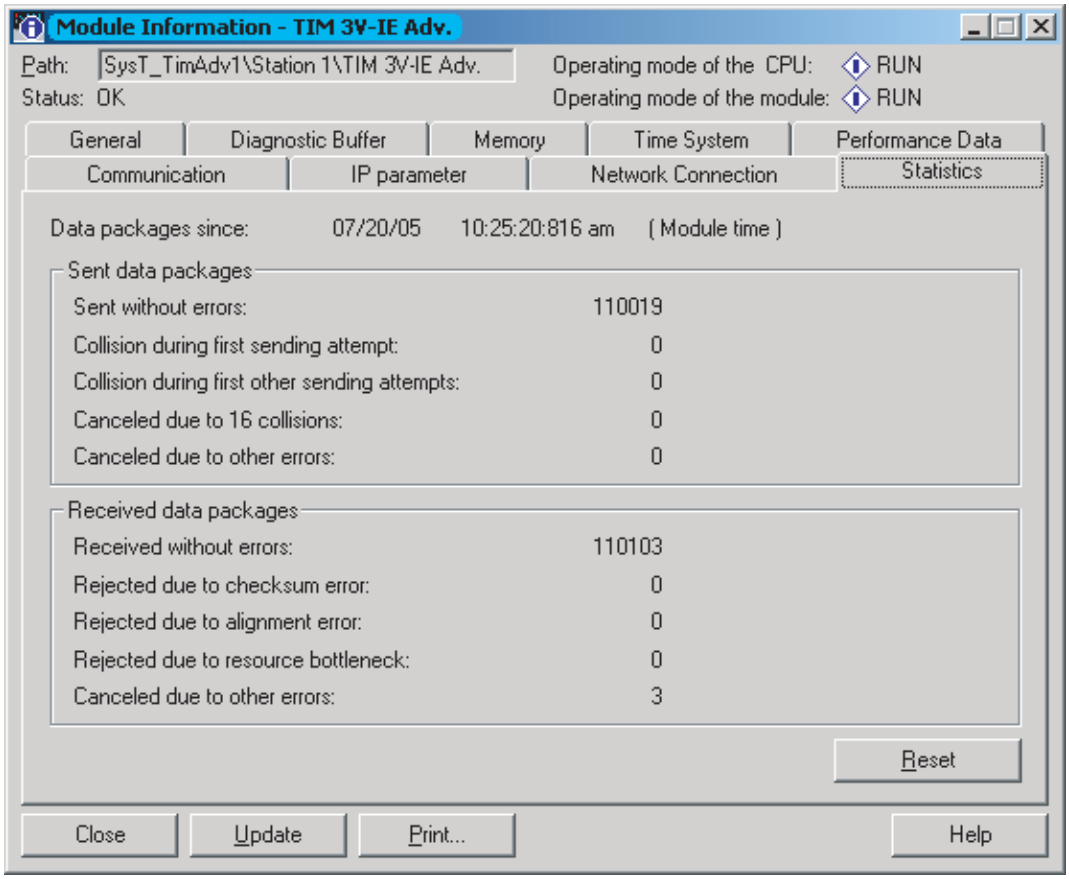


Figure 3-7 *Module Information* dialog, *Statistics* tab

3.2.3 Operating mode

Description of the functions

With the *Operating mode* function, you can change the operating mode of TIM and CPU modules. Apart from the operating mode, the current keyswitch setting and the last operating mode are displayed for CPU modules.

With TIM and CPU modules, the operating mode can be changed from *Run* to *Stop* or from *Stop* to *Run*. Changing the operating mode from *Stop* to *Run* triggers a restart on the TIM module.

Operator activities

1. Select a subscriber in the *SINAUT subscriber list* of the open project or in *Accessible Nodes*.
2. Open the dialog by selecting the *STEP 7 Diagnostics / Operating Mode* menu.

3. Click on the *Stop* button to stop the module.
4. Click on the *Warm Restart* button to restart the module.
   A TIM goes through a warm restart after approximately 10 seconds.

3.2.4 Setting the time

Description of the functions

The *Set Time of Day* function is used to display and set the date and time of a module. It is possible to set the module time to the PG/PC time or to set an edited time.

CPU modules have a hardware clock.
TIM modules have a software clock in the operating system of the module.

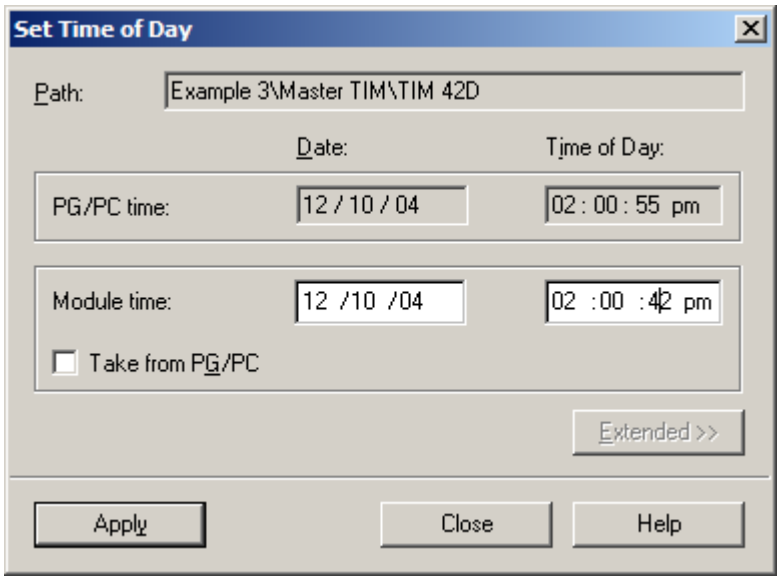


Figure 3-8 *Set Time of Day* dialog

Operator activities

1. Select a subscriber in the *SINAUT subscriber list* of the open project or in *Accessible Nodes*.
2. Open the dialog by selecting the *STEP 7 Diagnostics / Set Time of Day* menu.
3. To set the module time manually, click in the date or time display with the mouse, change the data and/or time using the keyboard and confirm by clicking the *Apply* button
   or or select the *Apply from PG/PC* option in the *Module time* field and confirm by clicking the *Apply* button.

3.3 SINAUT diagnostics

3.3.1 TIM Diagnostics

Description of the functions

The *TIM Diagnostics* function provides various diagnostic data of a TIM module. This is displayed in the following tabs:

- *Memory* tab:
  Information on memory and disk configuration
- *Message buffer* tab:
  Information on the buffer areas of an Ethernet TIM for messages
- *Communication* tab:
  Displays the installed communication drivers on the various interfaces of the TIM
- *Time synchronization* tab:
  Status of the time-of-day synchronization on the interfaces of the TIM
- *Time* tab:
  Information on the system clock of the TIM
- *Filesystem*:
  Displays all the files in the flash file system or (if installed) on the RAM disk of the TIM
- *IP Parameters* tab (TIM 4R-IE only):
  Displays the current IP parameters and settings of the Ethernet ports
- *Statistics* tab (TIM 4R-IE only):
  Displays the transmission statistics for the Ethernet ports

Operator activities

1. Select a subscriber in the *SINAUT subscriber list* of the open project or in *Accessible Nodes*.
2. Open the dialog by selecting the *SINAUT / TIM Diagnostics* menu.
3. Select the individual tabs with the mouse.
4. To display the interface-specific diagnostic data in the *Communication* and *Time Synchronization* tabs, select the name of an interface. The information on the relevant interface is displayed in the fields in the lower part of the two tabs.
   - In the *Communication* tab: Select an interface in the *Communication drivers* list box.
   - In the *Time Synchronization* tab: Select an interface in the *Communication interfaces for time synchronization* list box

Memory tab

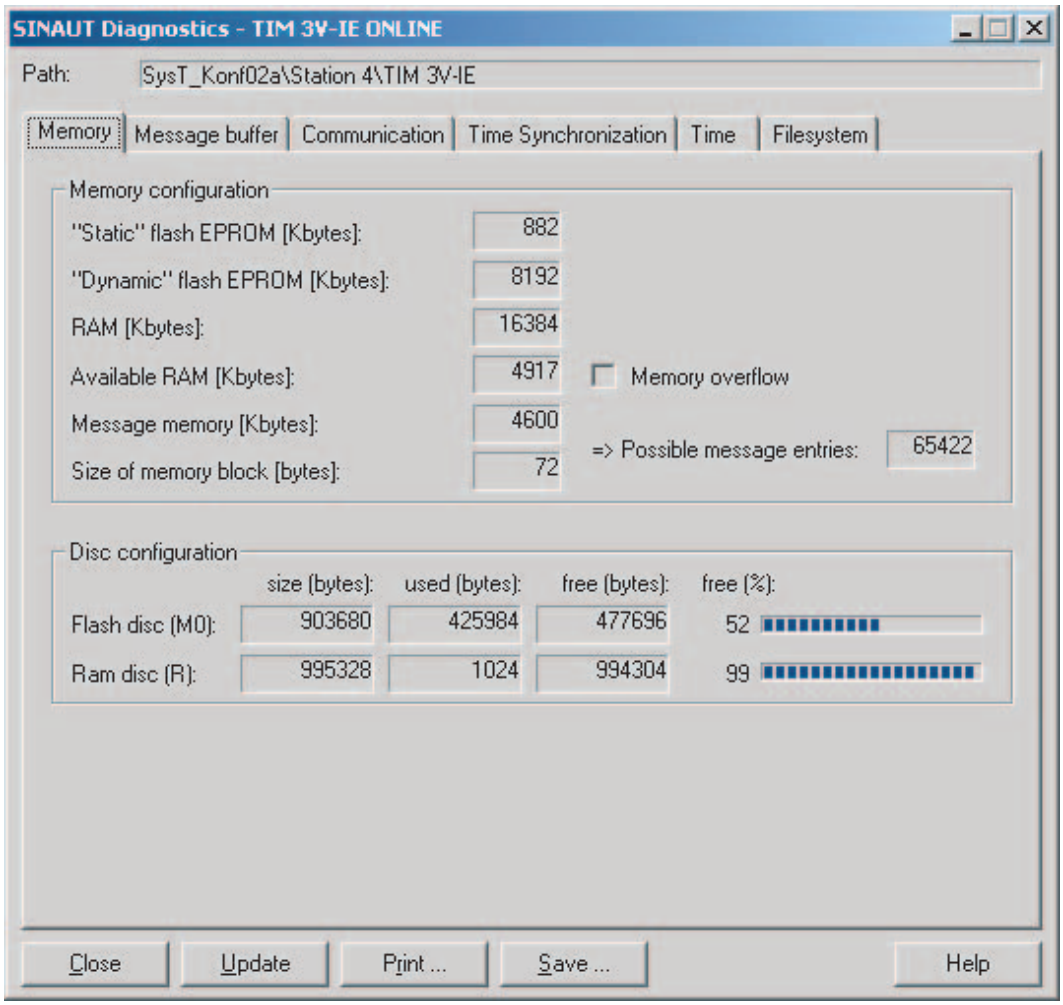


Figure 3-9 *SINAUT Diagnostics* dialog, *Memory* tab

The *Memory* tab displays current diagnostic data on the memory configuration on the TIM:

- In the *Memory configuration* field:
  - Static flash EPROM
  - Dynamic flash EPROM
  - RAM
  - Available RAM Size of the free storage space available to the drivers on the TIM for dynamic data.
  - Memory overflow: If the free RAM is no longer adequate, a checkmark appears in the *Memory overflow* check box.
  - Message memory: Size of the memory for data messages that can be stored
  - Size of a memory block that is reserved for a data message.
  - Number of possible message entries calculated based on the size of the message memory and the size of a memory block
- In the *Disc configuration* field:
  - Storage space, used and free storage capacity of the flash or RAM disk.

Message buffer tab

The *Message buffer* tab is available only for TD7onTIM-compliant TIM modules (for example TIM 3V-IE). The tab displays the current diagnostic data on the size and utilization of the buffer areas for messages on the selected TIM module.

With TD7onTIM-compliant TIM modules, the message buffer is divided into various buffer areas:

- Buffer areas for data messages sorted according to destination subscribers (destination subscriber buffers)
- Buffer areas for messages in TIM - TIM communication (TIM buffers)
- Buffer area for local communication with the CPU or an ST7cc/sc

The *Message buffer* diagnostic function analyzes the buffer areas of the destination subscribers in which the data messages are stored. These are of particular interest to the user.

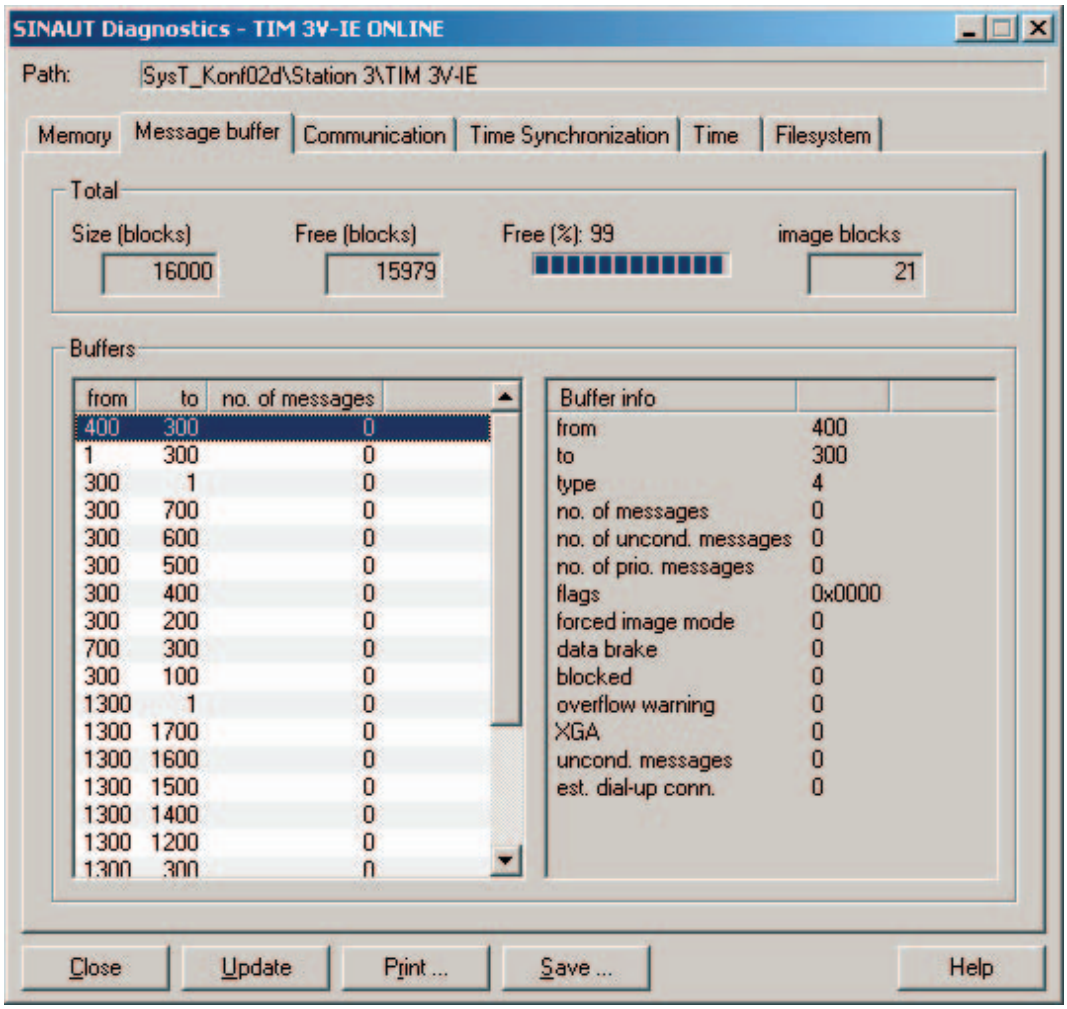


Figure 3-10 TIM Diagnostics - *Message buffer* tab

The *Total* output box provides the following information:

- *Size (blocks)*:
  Total size of the message buffer. The value indicates the total number of message memory blocks. This is calculated from the total size of the message memory set for the TIM and the byte size of a memory block. The parameter assignment is made in the network configuration in *NetPro* in the *Properties* dialog, *Options* tab, *Global message memory* field.
- *Free (blocks)*:
  Free area of the message buffer. The value indicates the number of free memory blocks.
- *Free (%)*:
  Free area of the message buffer as a percentage
- *image blocks*:
  Number of blocks occupied by the TIM in the message memory for data messages transmitted using the image memory principle.

The *Buffers* list box shows the message buffers for various communication partners with the following information:

- *from*:
  Subscriber number of the source subscriber
- *to*:
  Subscriber number of the destination subscriber
- *no. of messages*:
  Total number of stored messages for the relevant source and destination subscriber

If a message buffer is selected on the left with the mouse in the *Buffers* field, the following detailed information is displayed in the *Buffer info* list:

- *from*:
  Subscriber number of the source subscriber
- *to*:
  Subscriber number of the destination subscriber
- Type:
  - = 2: Buffer for organizational messages, hand-shake messages or message transmitted from one TIM to another TIM.
  - = 4: Buffer for messages to a remote subscriber (CPU or ST7cc).
  - = 8: Buffer for messages to a local subscriber (CPU or ST7cc).
- *no. of messages*:
  Total number of stored messages for the source and destination subscriber named at the top
- *no. of uncond. messages*:
  Number of stored messages to be sent unconditionally and spontaneously (only relevant in dial-up networks)
- *no of prio. messages*:
  Number of stored messages to be sent with high priority.
- *flags:*
  Flags is a hexadecimal value that codes the buffer information following it into binary.
- *forced image mode*:
  All data messages are currently processed in forced image mode; in other words, even send buffer messages are handled like image messages
- *data brake*:
  The sending of messages to the remote partner is currently disabled either because the remote partner is unavailable or there is a lack of memory on the remote partner.
- *blocked:*
  Reserved for future functions. Nothing is currently displayed.
- *overflow warning:*
  Reserved for future functions. Nothing is currently displayed.
- *XGA:*
  Reserved for future functions. Nothing is currently displayed.

- *uncond. messages:*
  Reserved for future functions. Nothing is currently displayed.
- *est. dial-up conn.*
  Reserved for future functions. Nothing is currently displayed.

Communication tab

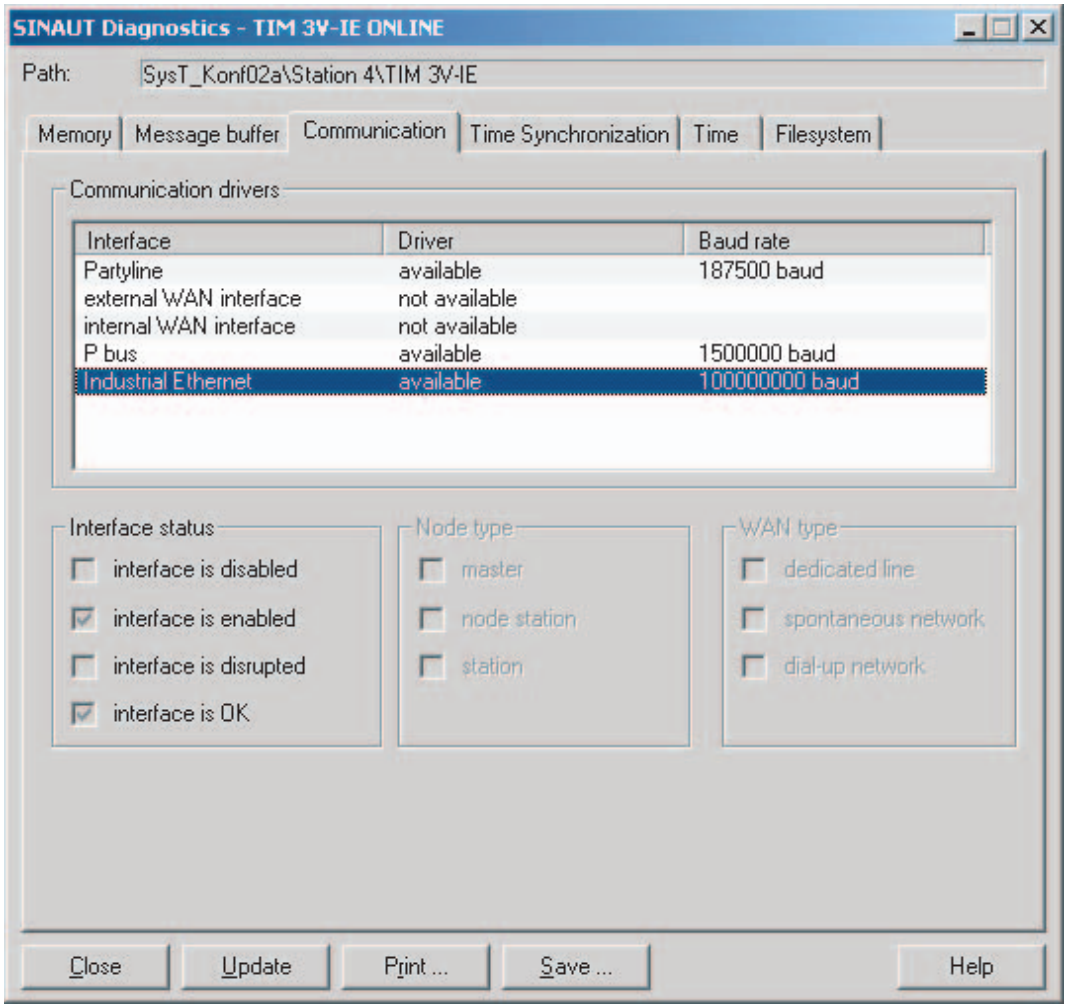


Figure 3-11 *SINAUT Diagnostics* dialog, *Communication* tab

The *Communication* tab displays information on the status of communication of the TIM with information on interfaces, drivers (available/not available) and baud rate. The data is displayed in the lower part of the dialog when you select one of the communication interfaces.

Time Synchronization tab

The *Time Synchronization* tab displays information on the time synchronization on the various interfaces of the TIM with information on the interface, synchronization and status of time synchronization. The information is displayed in the lower part of the dialog when you select one of the communication interfaces.

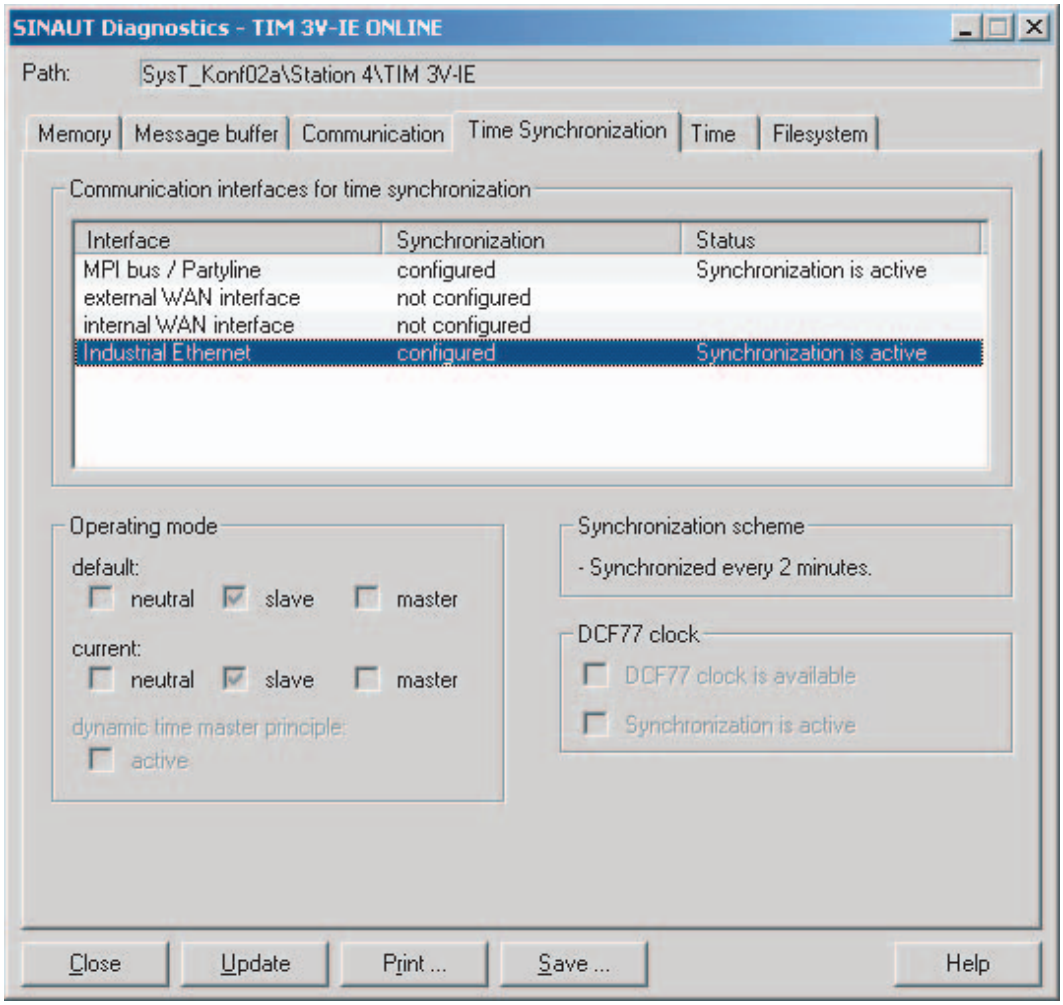


Figure 3-12 *SINAUT Diagnostics* dialog, *Time Synchronization* tab

Time tab

The *Time* tab displays the data and current module time of the TIM on the left in the *Current time* area. On the right in the *Clock status* area, information on the validity of the time, daylight saving/standard time and the changeover from daylight saving to standard time is displayed.

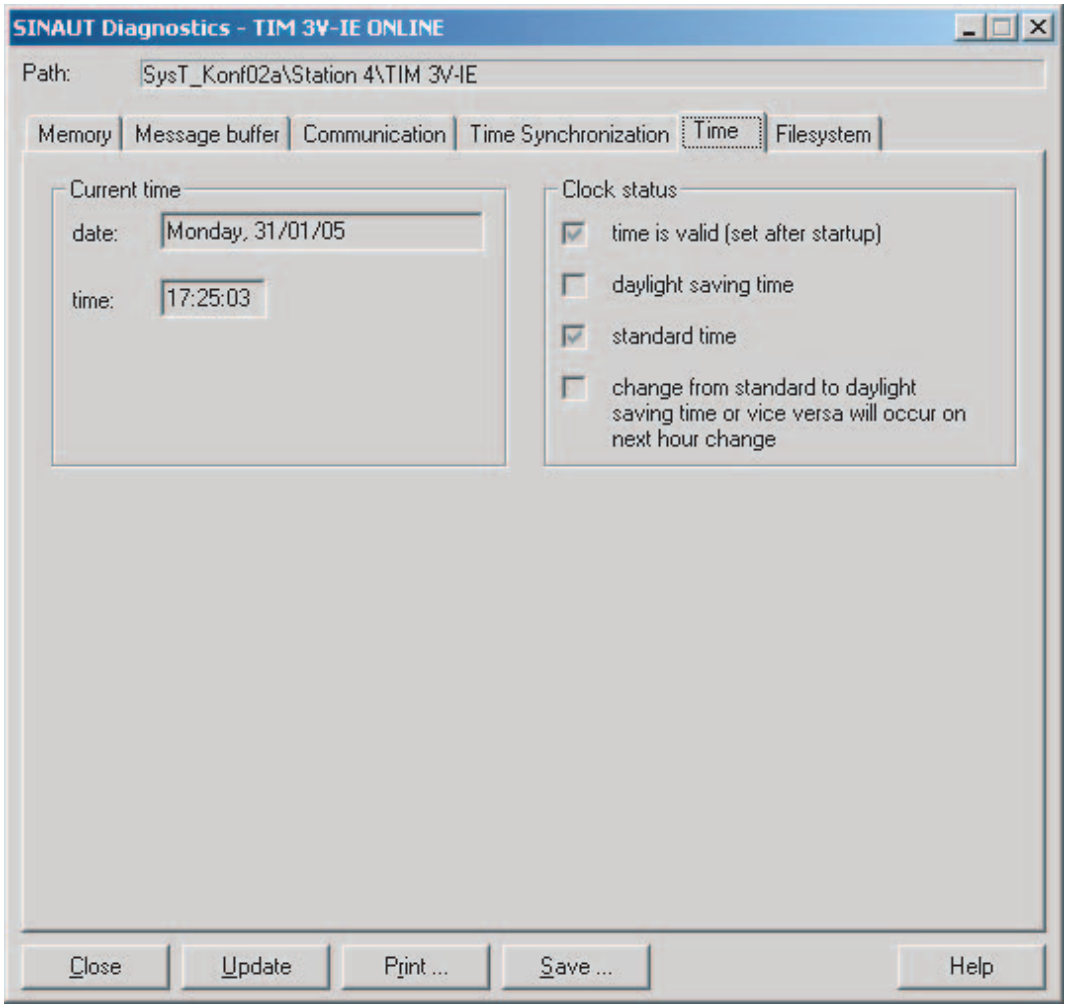


Figure 3-13 *SINAUT Diagnostics* dialog, Time tab

Filesystem tab

The *Filesystem* town displays all the system data blocks and files of the individual firmware components installed on the flash file system.

If a RAM disk is configured on the TIM, this is also displayed with the files it contains.

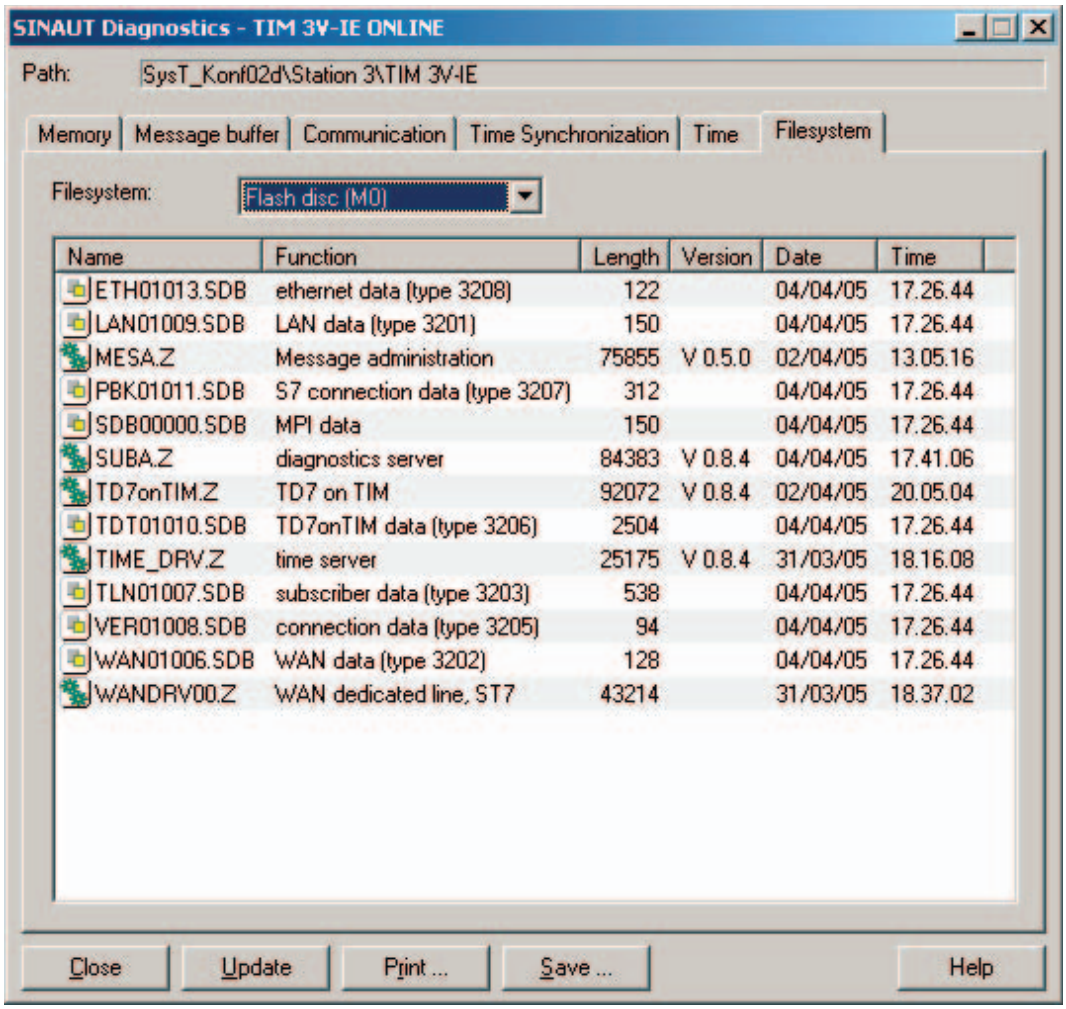


Figure 3-14 *SINAUT Diagnostics* dialog, Filesystem tab

IP Parameter tab

This tab is available only for the TIM 4R-IE.

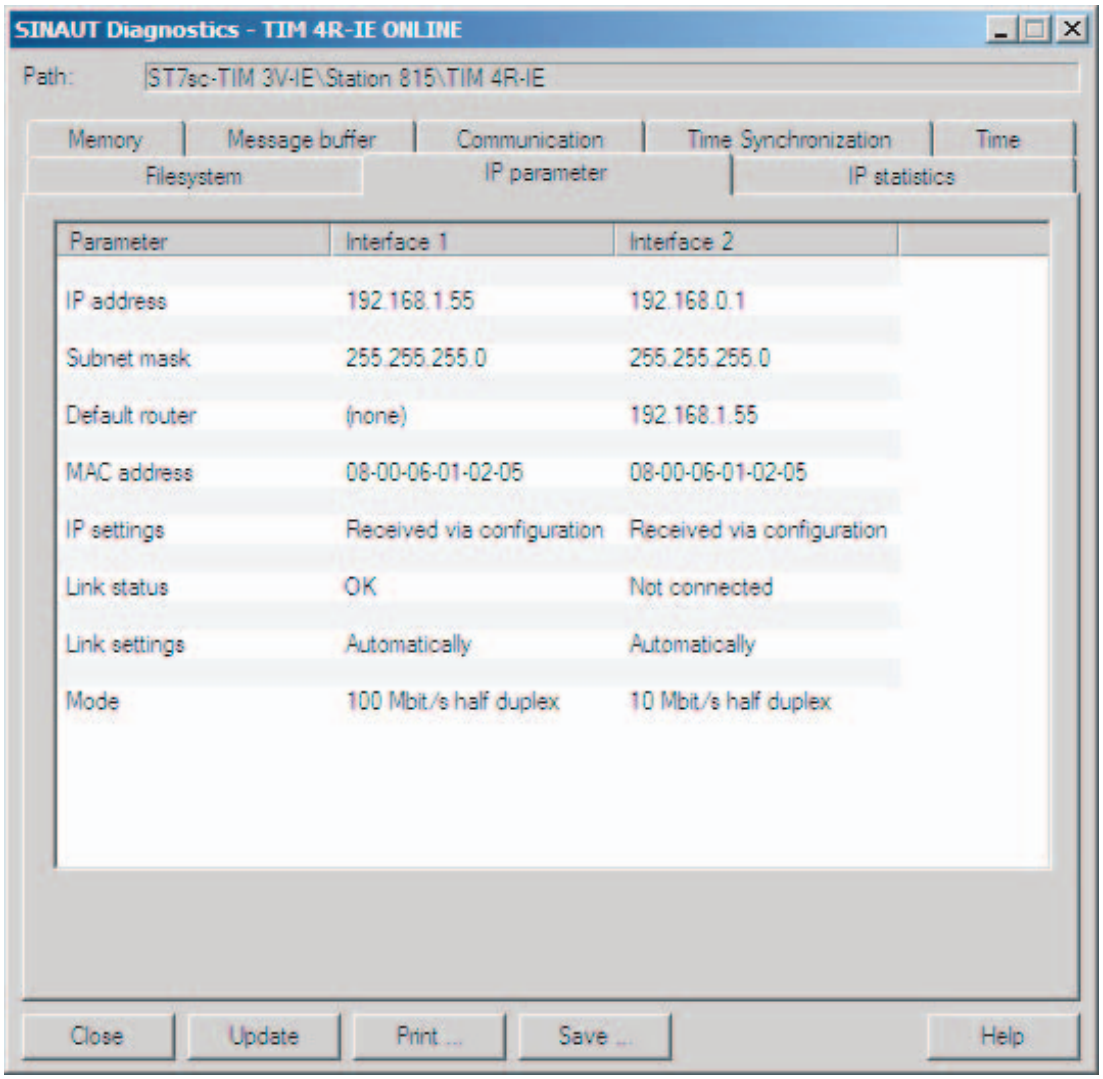


Figure 3-15 *SINAUT Diagnostics* dialog, *IP Parameters* tab

The *IP Parameters* tab displays the current IP parameters and settings of the Ethernet ports:

- IP Address: Configured IP address of the module.
- Subnet Mask: Configured subnet mask of the module.
- Default router: If a default router was specified during configuration, the IP address of the default router is displayed here.
- MAC address: MAC address of the module
- IP settings: Indicates where the module obtained the IP parameters from.
- Link Status: Indicates whether or not a physical connection to Ethernet exists.

- Link settings: Shows the setting for detecting network settings, here: "Automatic" (Autosensing)
- Mode: Indicates the transmission speed and duplexity on Ethernet.

Statistics tab

This tab is available only for the TIM 4R-IE.

The *Statistics* tab contains transmission statistics for the Ethernet ports. The number of transferred data packets with and without errors since the last reset or restart of the module is displayed for the send and receive directions.

MSC Communication tab

This tab exists only for a TIM 4R-IE as MSC master station.

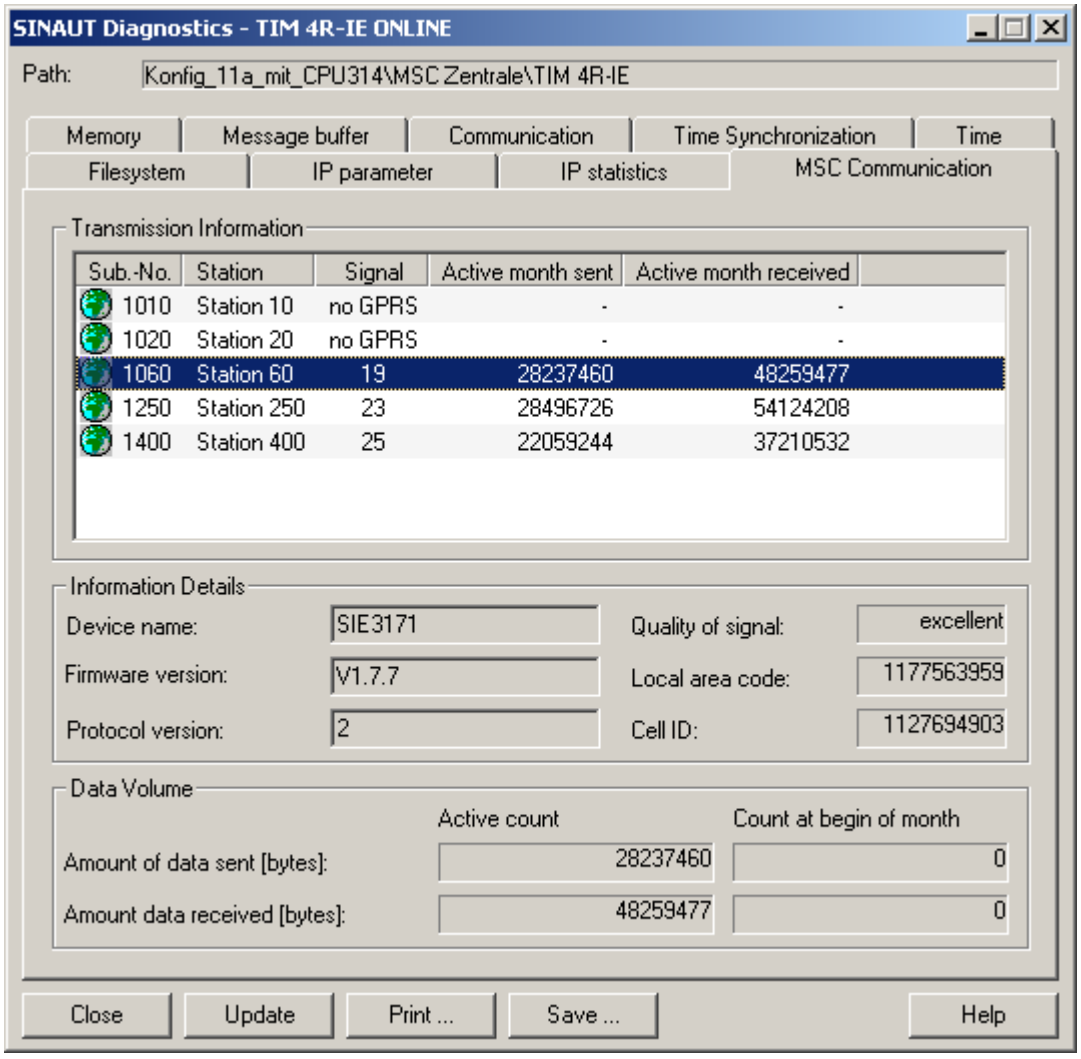


Figure 3-16 *SINAUT Diagnostics* dialog, *MSC Communication* tab

The "Transmission Information" list shows all MSC stations along with the following detailed information stored on the MSC master when the dialog is opened.

- Globe icon

  The globe icon in this dialog has a the following meaning depending on its color (from the perspective of the master station):

  – Black and white ( )

    The station can be reached online but the displayed data is not up to date. The reason for this may be that the response to a query was not received, for example due to a timeout. *)

  – Green ( )

    All the required properties of the provider are configured. The provider can be reached online.

  – Blue and red ( )

    The provider cannot be reached.

  *) When the "MSC Communication" tab is opened, the icons of all stations are gray. The change to green or blue and red takes place only after the data has been updated (see below).

- Subscriber number
- Station
- Signal quality

  see below → "Quality of the received signal"

- Sent current month / Received current month

  Sent/received volume of data since the last change of month

  Difference between "current count - count previous month" (see below)

  – When the TIM of a DSL station is restarted, the counter is set to 0.

  – When the MD720-3 modem of a GPRS station is restarted, the counter is set to 0.

The fields below this show more details of the MSC station selected in the list:

- Device name

  Device name of the TIM

- Firmware version

  TIM firmware version

- Protocol version

  MSC protocol version

The following information is only supplied by GPRS stations with MD720-3:

- Signal quality

  Relative receive quality
  Range: 0...99

0 = error-free signal
99 = loss of synchronization

- Local area code

  Location of area in the mobile wireless network

- Cell ID

  ID number of the active GSM cell

- Data sent [bytes] / Data received [bytes]

  Volumes of data in bytes at the following times:

  – Current count

    Current counter reading: Volume of data since the first station TIM startup to the present time

  – Count in previous month

    Volume of data at the last change of month (1st day of the month)

Buttons in the *MSC Communication* tab

- Update

  The "Update" button updates the data of the stations as follows:

  – No station selected: The data of all stations is updated.

  – One station selected: The data of the selected station is updated.

  The requested data can take up to 30 seconds.

- Print...

  Opens a dialog from which you can print the displayed information.

- Save...

  Opens a dialog from which you can save the displayed information in a text file.

3.3.2 TIM subscriber diagnostics

Description of the functions

The *TIM Subscriber Diagnostics* function displays the diagnostic data of the known SINAUT subscribers of the connected TIM module. The following detailed information is available:

- Selection list of the known subscribers (on the left):
  The selection list of the known subscribers is used to select individual subscribers known to the connected TIM module allowing the information to be displayed in the tabs on the right. The known subscribers are listed with their subscriber number and subscriber type, if accessed over a STEP 7 project the name and station is also displayed.

- *Status* tab:
  Information on the availability of the partners or connection disruptions and information on data communication and the operation of the send buffer

- *Partner* tab:
  Displays the known partners:
  - With CPU modules: Display in the known partners tab of the CPU selected in the list of known subscribers on the left of the dialog
  - With TIM modules: Display in the known partners tab of the connected TIM module
- *Dialing extern* (optional):
  Dial-up service and command of the connected TIM module
- *Polling intern* (optional):
  Data of the station poll of the partners of the connected TIM module connected over a dedicated line with the option of disabling or enabling a connected partner

The colored symbols in the selection list of known subscribers indicate the availability of the individual subscriber and have the following meaning:

Table 3- 2 Symbols indicating subscriber availability in subscriber diagnostics

Symbol	Status
	Subscriber is available, all connections OK
	Subscriber is available, at least one connection is disrupted
	Subscriber is not available

Operator activities

1. Select a TIM module in the *SINAUT subscriber list* of the open project or in *Accessible Nodes*.
2. Open the dialog by selecting the *SINAUT / TIM Subscriber Diagnostics* menu.
3. Select a subscriber in the selection list of the known subscribers on the left in the dialog.
4. To display the information on *Status*, *Partner*, *Dialing extern* or *Polling intern*, click on the relevant tab.

Status tab

The *Status* displays the following information on the subscriber selected on the left in the list of known subscribers from the perspective of the connected TIM module:

- In the *General* field:
  - Availability of the known subscriber. Entries indicating problems are highlighted.
  - Any connection disruptions
  - Information on gateways to the known subscriber (subscriber local / remote)
- In the *Connection* field:
  - Interface of the connection
  - Type of connection
  - Connection enable
  - Information on polling
  - Status of data communication
- In the *Special* field:
  - Information on the operation of the send buffer of the known subscriber

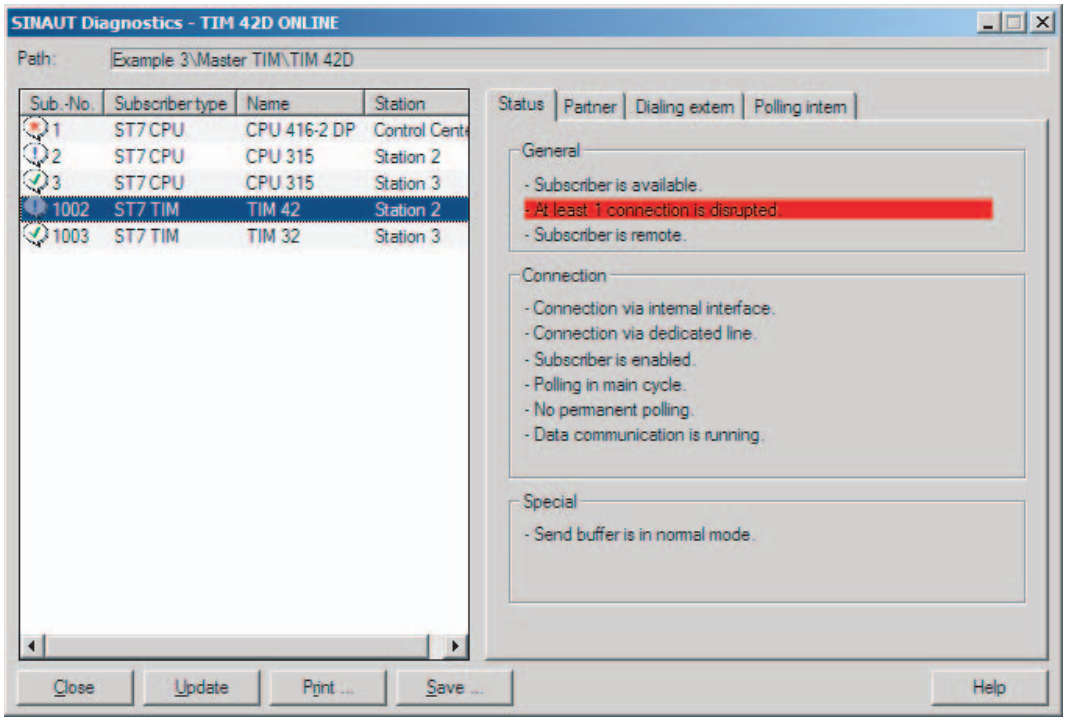


Figure 3-17 *TIM Subscriber Diagnostics* dialog, *Status* tab

Partner tab

The *Partner* tab displays the following no partners with their subscriber number, name and station in the *List of partners* area:

- With CPU modules: Display of the communication partners of the CPU selected in the list on the left of the dialog
- With TIM modules: Display of the communication partners of the connected TIM module selected in the subscriber list prior to opening the dialog.
  If different subscribers are selected on the left in the dialog, the same subscribers are always displayed in the *List of partners*.

With TIM modules, partners are only displayed in the TIM is installed in the master station and is connected to partners over a dedicated line.

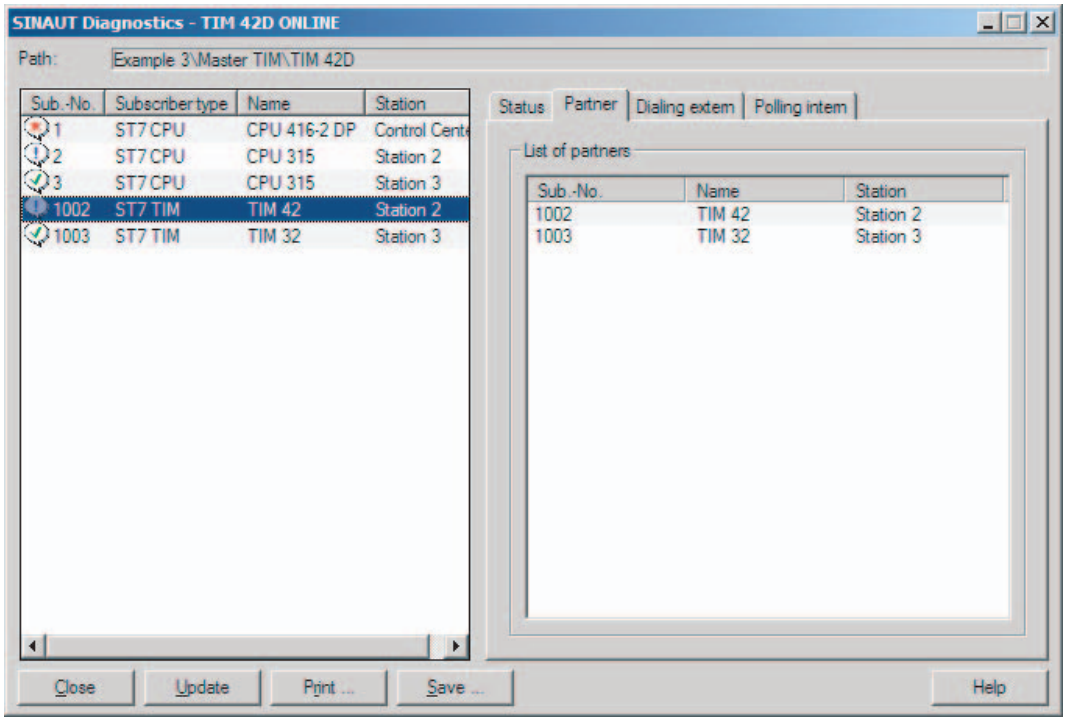


Figure 3-18 *TIM Subscriber Diagnostics* dialog, *Partner* tab

Dialing extern tab

If a TIM with master functionality is selected *before* you open the dialog, the *Dialing extern* tab displays the special services and the telephone number list (dial command) of the dial-up network driver of the connected TIM modules.

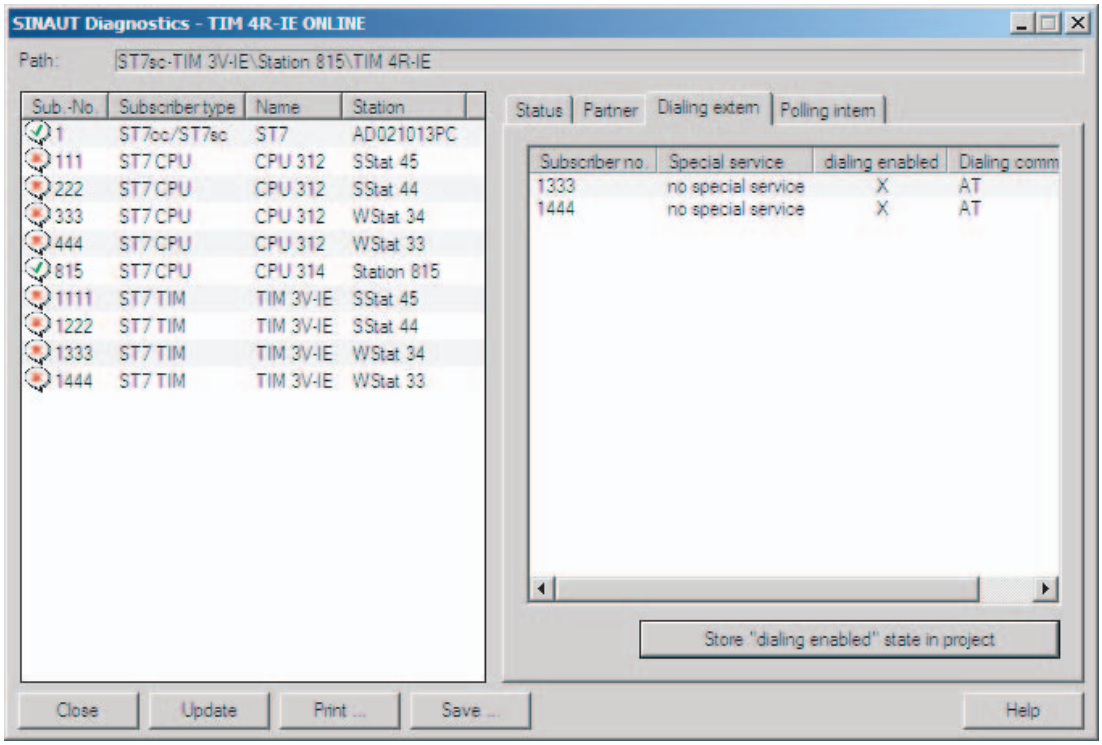


Figure 3-19 *TIM Subscriber Diagnostics* dialog, *Dialing extern* tab

The parameters *Subscriber number*, *Special service*, *Call enabled* and *Dial command* are displayed for the listed TIM modules.

An "X" in the *Call enabled* column means that the connection is enabled.

The following functions are available only for the TIM 4R-IE:

- By double-clicking on a subscriber in the dialing list, the *Disable / Enable Subscribers* dialog opens in which the configured and current enable states of the selected subscriber are displayed from the perspective of the master TIM. The current enable status can be changed.
- The settings for enabling subscribers are stored permanently.
- Using the button below the list, you store the current enable status in the STEP 7 project of the connected PG/PC.

Polling intern tab

If a TIM with master functionality is selected *before* opening the dialog, the *Polling intern* tab displays the data of the station for for the known subscriber connected to the connected TIM over a dedicated line.

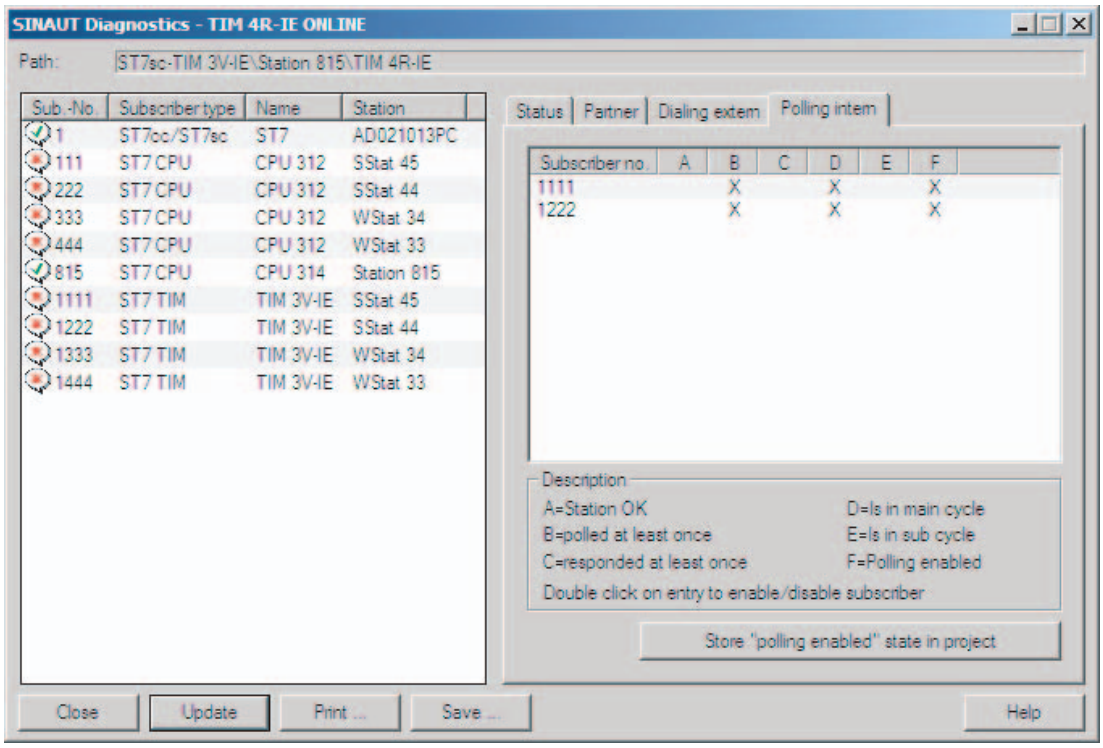


Figure 3-20 *TIM Subscriber Diagnostics* dialog, *Polling intern* tab

By double-clicking on a subscriber in the polling list, the *Disable / Enable Subscribers* dialog opens in which the configured and current enable states of the selected subscriber are displayed from the perspective of the master TIM. The current enable status can be changed.
If you change the status to *Disable* in this dialog, the configured status is adopted again next time the master TIM is reset.

The following functions are available only for the TIM 4R-IE:

- The settings for enabling subscribers are stored permanently.
- Using the button below the list, you store the current enable status in the STEP 7 project of the connected PG/PC.

3.3.3 TIM diagnostic messages

Functional description

With the *TIM Diagnostic Messages* function, **Extended diagnostic messages** are activated or deactivated and selected levels for various components of the TIM firmware. The extended

diagnostic messages contain detailed information on subfunctions of individual firmware components and are entered in the diagnostic buffer of the TIM.

The extended diagnostic messages are displayed in hexadecimal format.

The display of extended diagnostic messages can be selected for various firmware components and some functions (diagnostic areas). The diagnostic areas of the TIM 3 / TIM 4 and the Ethernet TIMs are different:

Table 3- 3 Diagnostic areas of the TIM 3 / TIM 4

Firmware section of the TIM	Diagnostic area	Meaning / subfunction
Diagnostic server		
	1	Reception of organizational messages
	2	Management of organizational messages
	3	System status list query
	4	Data synchronization with driver and device redundancy
	5	Not used
Routing server		
	1	not assigned
	2	not assigned
	3	not assigned
	4	not assigned
	5	not assigned
	6	not assigned
Installation program		
	1	not assigned
	2	not assigned
Clock driver		
	1	WAN/LAN synchronization cycle
	2	not assigned
	3	not assigned
	4	not assigned
External WAN driver		
	1	Interrupt level (receive direction) error messages
	2	Interrupt level (receive direction) level 1
	3	Interrupt level (receive direction) level 2
	4	Control level
	5	Task management in WAN driver
	6	not assigned
	7	not assigned
Internal WAN driver		
	1	Interrupt level (receive direction) error messages
	2	Interrupt level (receive direction) level 1
	3	Interrupt level (receive direction) level 2
	4	Control level

Firmware section of the TIM	Diagnostic area	Meaning / subfunction
	5	Task management in WAN driver
	6	not assigned
	7	not assigned

Table 3- 4 Diagnostic areas of the Ethernet TIMs

Firmware section of the TIM	Diagnostic area	Meaning / subfunction
Start manager		
	1 - 8	Reserved
P bus driver		
	1 - 8	Reserved
Clock driver		
	1 - 8	Reserved
LAN communication		
	1 - 32	Reserved
LAN communication		
	1 - 32	Reserved
Subscriber administration		
	1 - 32	Reserved
Message buffer		
	1 - 32	Reserved
TD7onTIM		
	1 - 32	Reserved
WAN driver 1		
	1 - 32	Reserved

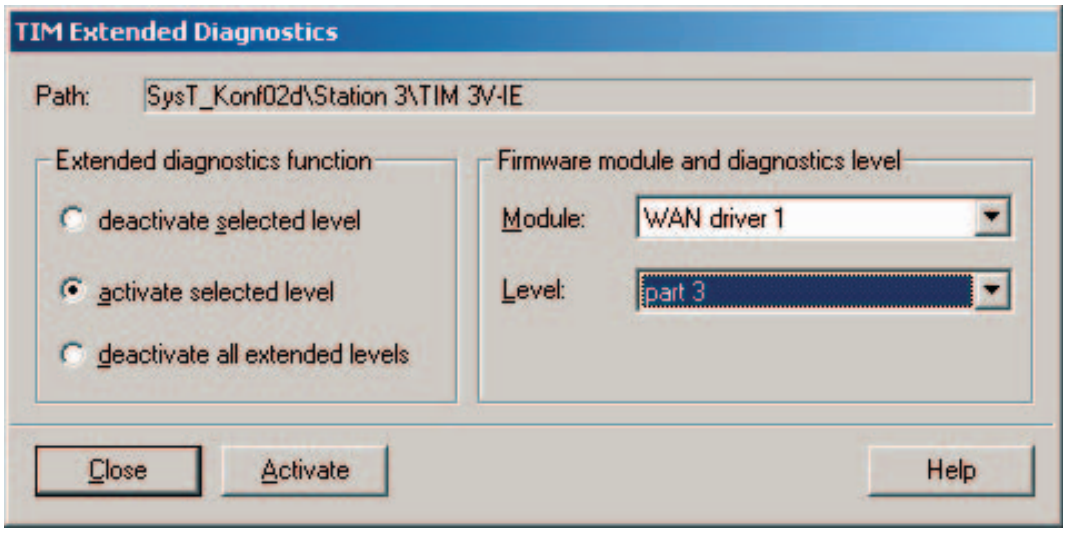


Figure 3-21 *TIM Extended Diagnostics* dialog

Operator activities

1. Select a subscriber in the *SINAUT subscriber list* of the open project.
2. First open the *Module Information / Diagnostic Buffer tab* by selecting the *STEP 7 Diagnostics/ Module Information / Diagnostic Buffer tab* and click on the *Settings* button.
3. Make sure that the *Update display during operating mode transition* option is deselected (no check mark) at the bottom of the *Settings for Display Diagnostic Buffer* dialog and confirm with *OK*. You can leave the *Module Information* dialog open.
4. Change to the SINAUT Diagnostics and Service tool and open the *TIM Extended Diagnostics* dialog by selecting the *SINAUT / TIM Diagnostics* menu.
5. Select the *required function* in the field on the left of the dialog.
6. Then select the following in the *Firmware module and diagnostics level* area
   - The required firmware component in the *Module* list box and
   - The required level (area) in the *Level* list box.
7. Confirm your entries by clicking on the *Activate* button. A dialog *Loading* opens briefly and indicates that the activation information for extended diagnostics is being sent to the module by displaying a progress bar. Once the information has been sent successful, the *Loading* and *TIM Extended Diagnostics* dialogs are closed.
   Any diagnostic messages are activated on the selected module and displayed in the active diagnostic buffer.
   If multiple extended diagnostic messages are activated, you must confirm the activation of the message output for each individual firmware component and level with *Activate*.
8. Change back to the *Module Information / Diagnostic Buffer tab* that is still open and click on *Update*, if necessary, several times. Extended diagnostic messages are displayed in hexadecimal code.
   If necessary, save the diagnostic messages as a text file as described for the *Module Information* function.

9. To disable the output of extended diagnostic messages for an individual level, select the *deactivate selected level* option in the *TIM Extended Diagnostics* dialog and close the dialog with the *Activate* button.
10. To disable the output of all extended diagnostic messages for all firmware components and all levels of the selected subscriber, select the *deactivate all extended levels* option in the *TIM Extended Diagnostics* dialog and close the dialog with the *Activate* button.

NOTICE
If even one level of the extended diagnostic messages is activated, the size of the TIM diagnostic buffer is increased from 50 to 200 entries. Due to the increased memory requirements, extended diagnostic messages should not be activated permanently. After recording and saving the extended diagnostic messages, deactivate the *Extended Diagnostics* function again. The simplest method is to a deactivate the function with the *deactivate all extended levels* option. When you deactivate extended diagnostics, the size of the TIM diagnostic buffer is set back from 200 to the basic setting of 50 entries.

3.3.4 TIM Message Monitor

Description of the functions

The *TIM Message Monitor* function is used to specify the settings for message monitoring of a selected TIM and starts the monitoring.

In TIM message monitoring, the messages received and sent by the TIM are recorded. As soon as the monitoring function is activated, copies of every message are stored in a buffer created specifically for this function. The messages are read out of the buffer of the TIM and saved in a monitoring file.

To start to message monitoring, you must set the following:

- the output file in which the recorded messages are saved and
- the stop action for the monitoring, either
  - manual for
    - continuous sampling (reading out the buffer at 5 second intervals) or
    - read data once after stop of monitor,
  - fill buffer only once or
  - time-limited reading by specifying the elapsed time.

Reading the data once after stopping monitoring is set if the messages are to be monitored following any intervention, for example turning off the TIM.

If the message buffer of the TIM is read once without any time limitation, the entire saved SINAUT data traffic of the relevant TIM is read out. With the fixed buffer size of the TIM, this involves 400 messages.

Monitoring is always started manually in the *TIM Message Monitor* dialog. When monitoring is started, the *TIM Message Monitor* progress bar is displayed and the group error LED of the TIM flashes while monitoring is active.

If the *manual* stop option is selected, monitoring is also stopped in the *TIM Message Monitor* progress bar.

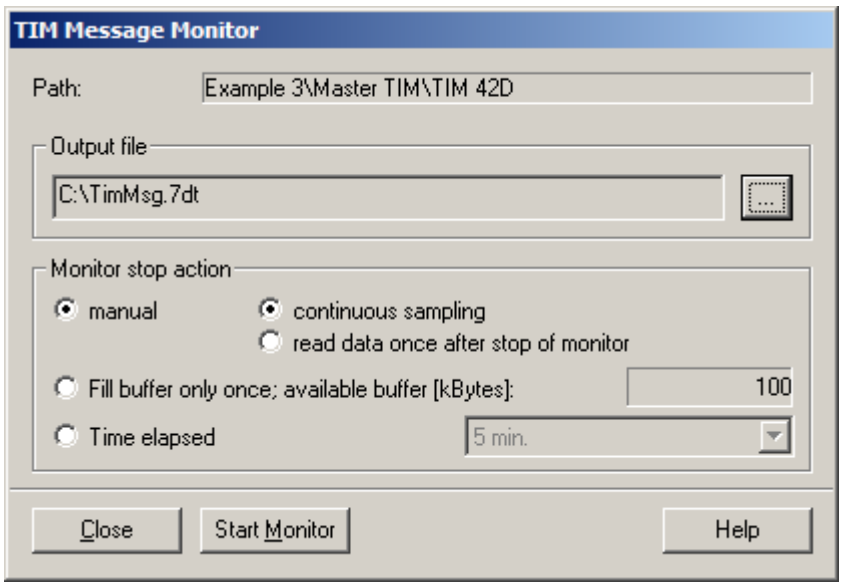


Figure 3-22 *TIM Message Monitor* dialog

Operator activities

1. Select a subscriber in the *SINAUT subscriber list* of the open project.
2. Open the *TIM Message Monitor* dialog by selecting the *SINAUT / TIM Message Monitor* menu.
3. In the *Output file* box, enter the name of a file of the type **.7dt*, in which your recorded TIM messages will be saved or browse for the directory of the 7dt file in the file tree using the square button.
4. In the *TIM Message Monitor* dialog, in the *Monitor stop action* area, select one of the three options for starting message monitoring. If you select manual, you must decide whether the recorded messages are read by continuous sampling or after stopping monitoring.
5. Start monitoring with the *Start Monitor* button. The *TIM Message Monitor* progress dialog opens and displays information on the acquisition mode and the progress of the message recording over time indicating the elapsed and remaining time. The amount of data read and data remaining is also displayed.
6. Click on the *End Monitor* button in the *TIM Message Monitor* dialog to stop monitoring manually.
7. In the next dialog, decide whether you want to read the monitored messages immediately (the *TIM Message protocol* window opens) or whether you want to read the monitored messages at a later point in time (you return to the SINAUT subscriber list).

Note

You can open and evaluate the TIM message protocol later using the *TIM message protocol* function of the Diagnostics and Service tool.

3.3.5 TD7 CPU Diagnostics

Description of the functions

The *TD7 CPU Diagnostics* function displays all the diagnostic messages in the diagnostic buffer of a previously selected CPU. As default, this is 10 messages for the CPU. The display in this dialog is for*:*

- Diagnostic messages generated by SINAUT TD7 as plain text
- All other STEP 7 diagnostic messages in hexadecimal code.

The message list at the top of the dialog contains the diagnostic messages with information on the message number, time, date and event. The entries are sorted in descending chronological order; in other words, the latest message is at the top.

In the lower part of the dialog in *Details of event*, you can see the event ID and additional information on the message selected above in the message list.

Note

If you also want to view the STEP 7 diagnostic messages in plain text as well as the messages generated by TD7, open the *STEP 7 Diagnostics / Module Information - Diagnostic Buffer tab* dialog and position this next to the *TD7 CPU Diagnostics* dialog that is already open.
The STEP 7 diagnostic messages displayed in hexadecimal format in the *TD7 CPU Diagnostics* dialog are then displayed alongside as plain text in the *STEP 7 Diagnostics / Module Information* dialog.

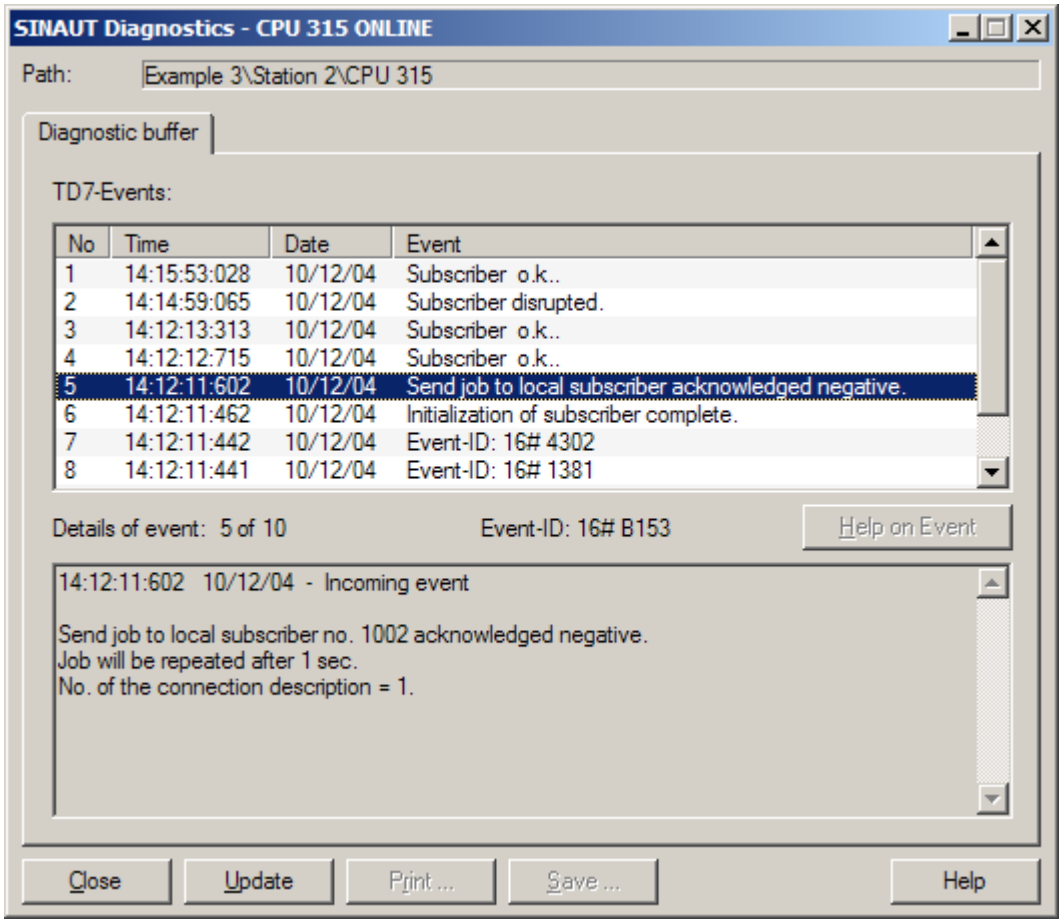


Figure 3-23 *SINAUT Diagnostics* dialog of the *TD7 CPU Diagnostics* function

Operator activities

1. Select a CPU module in the *SINAUT subscriber list* of the open project.
2. Open the dialog by selecting the *SINAUT / TD7 CPU Diagnostics* menu.
3. Select a message in the list, the additional information on this message is displayed below the message list in the *Details of event* text box.
4. Click on *Update* to update the display in the dialog with the most recent messages from the diagnostic buffer.

3.3.6 TD7 Block Structure

Description of the functions

The *TD7 Block Structure* function provides information on the software blocks of a CPU. The function starts the block structure analysis of a previously selected CPU. It does not access the module but rather the project data.

The results are displayed in four tabs:

- *Statistics* tab:
  This indicates the number, name, type and station of the selected subscriber and displays the communication partner and number of objects of the communication links
- *Block tree* tab:
  This displays a tree structure with information on the paths and all SINAUT blocks and the data of all SINAUT block calls for the subscribers of the entire project
- *Block list* tab:
  Presents the data of all SINAUT block calls in the form of a list
- *Plausibility* tab:
  Shows the results of plausibility checks for the calls of the SINAUT blocks *BasicTask*, *Startup, Safe* and *ListGenerator* and for the unique assignment of DP numbers, partner numbers and partner object numbers

The tree structure of the subscriber can be saved in the form of an XML file in all tabs using the *Save* function.

The tree structure of a different version, a different subscriber or project previously saved as an XML file can be opened and viewed with the *Load* function in all tabs of the open dialog. The previous view is then overwritten.

Operator activities

1. Select a subscriber in the *SINAUT subscriber list* of the open project.
2. Open the dialog by selecting the *SINAUT / TD7 Block Structure* menu.
3. Select the relevant tab to display the information.

Statistics tab

The *Statistics* tab lists the number, name, type and station of the selected subscriber in the *Subscriber Information* area on the left.

The *Communication Links* area on the right displays the communication partners of the selected CPU and the number of communication objects per communication partner (target subscriber).

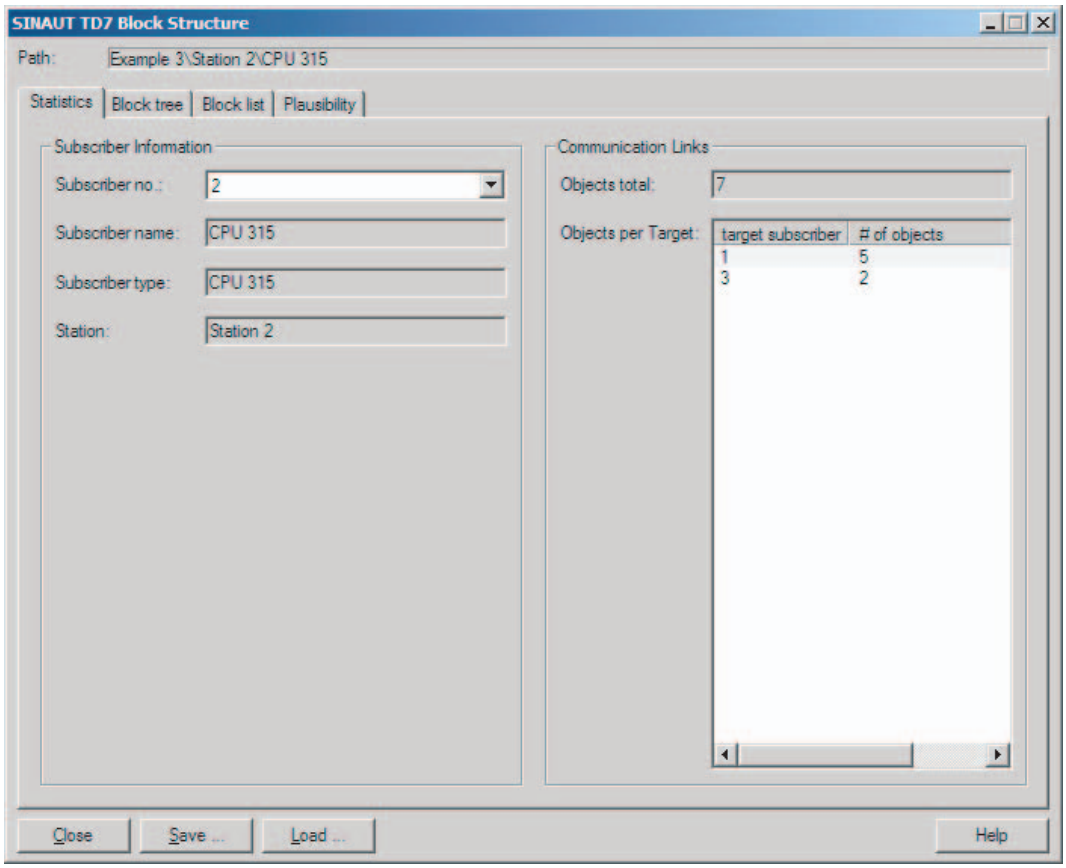


Figure 3-24 *SINAUT TD7 Block Structure* dialog, *Statistics* tab

Block tree tab

The *Block tree* tab shows the information obtained on the project path and the path to the previously selected subscriber in a tree structure. The following information is listed:

- Project Information (__*Info*)
  with general information on the project path and the logical path
- *Subscriber* number / *station* name
  with all system blocks and user objects of the selected subscriber:
  - All SINAUT data blocks (__*Blocks*) in the program directory of the CPU
  - Information (__Info) on the module name, station name, type name, and subnumber of the subscriber
  - The system blocks. These can be: BASICTASK, PARTNERMONITOR, SAFE, PARTNERSTATUS, PULSECOUNTER, STARTUP, TIMETASK, STARTUP
  - The user objects (analog value, binary value, command, counter, setpoint objects etc.) with
    - information on their call data and a
    - listing of the most important parameters of the individual user objects

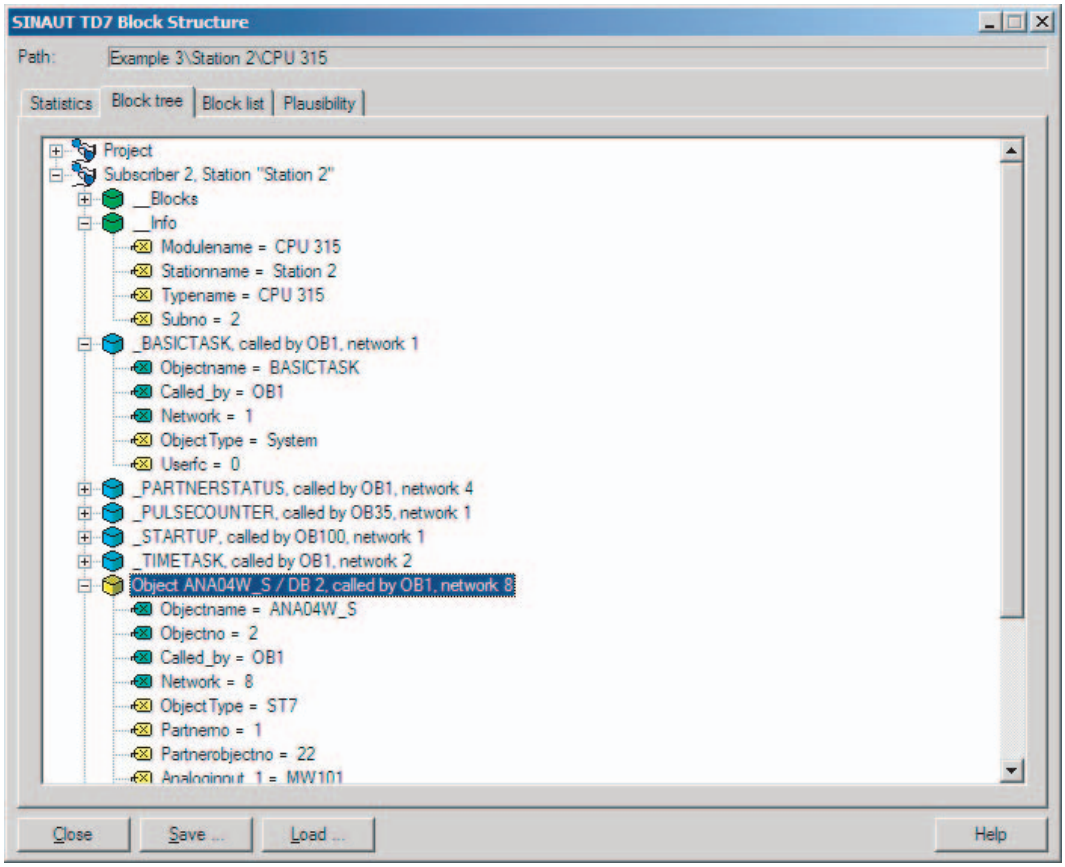


Figure 3-25 *SINAUT TD7 Block Structure* dialog, *Block tree* tab

Block list tab

The *Block list* tab presents the data of all SINAUT block calls in the form of a list. As default, the blocks are displayed in the following five columns:

- **Subscriber**: This contains the block icon and the subscriber number of the CPU. The system blocks have a blue icon, the user blocks a yellow icon.
- **Object name**: Name of the system block or the user object
- **Objectno.**: Object number of the SINAUT object, the number of the instance DB used on the CPU
- **Partnerno.**: Subscriber number of the SINAUT destination or source subscriber as part of the SINAUT addressing (subscriber number, object number)
- **Partnerobjectno.**: Destination or source object number as part of the SINAUT addressing

You can change the ascending order of the blocks within the block list by clicking on the column headers to sort according to the parameters of the individual columns.

Further parameters are added to the view of the block list as follows:

1. Right-click within the tab and click on the *Add columns* context menu.
2. In the *Add columns* dialog that opens, click on the required parameters and confirm with the *OK* button.

The added block parameters are not included permanently in the list view and are no longer present the next time you open the dialog.

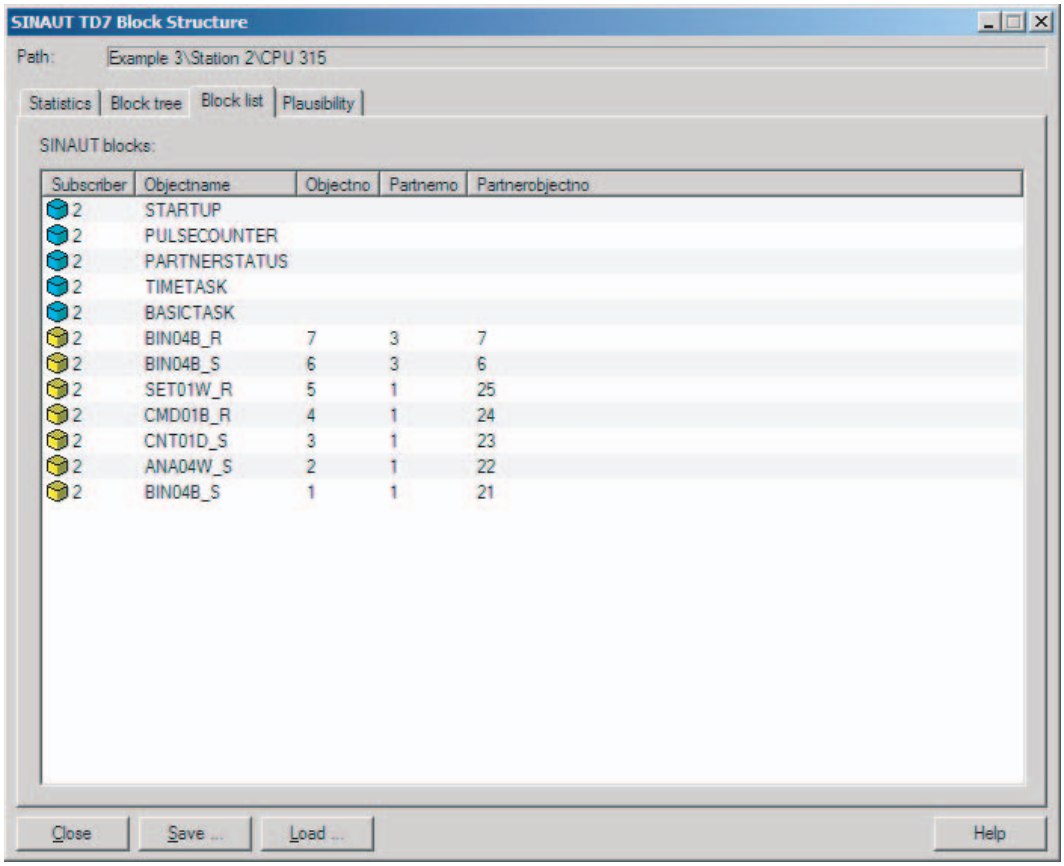


Figure 3-26 *SINAUT TD7 Block Structure* dialog, *Block list* tab

Note

The data of the block list are required for the configuration of a SINAUT ST7cc/sc control center and can be saved for this purpose in an XML file.

Plausibility tab

The *Plausibility* tab runs a series of CPU-specific plausibility checks for the selected subscriber and displays the *Check result*. The following is checked:

- Whether the *Startup* SINAUT block was called correctly,
- Whether the *Safe* SINAUT block needs to be called and was actually called,
- Whether the object numbers (DB numbers) of the SINAUT blocks were assigned uniquely,
- Whether the partner numbers and partner object numbers were assigned uniquely,
- Whether the *BasicTask* SINAUT block was called correctly,
- Whether the *ListGenerator* SINAUT block needs to be called and was actually called.

If errors occur in the plausibility checks listed above, a corresponding message is displayed in the *Check result* text box.

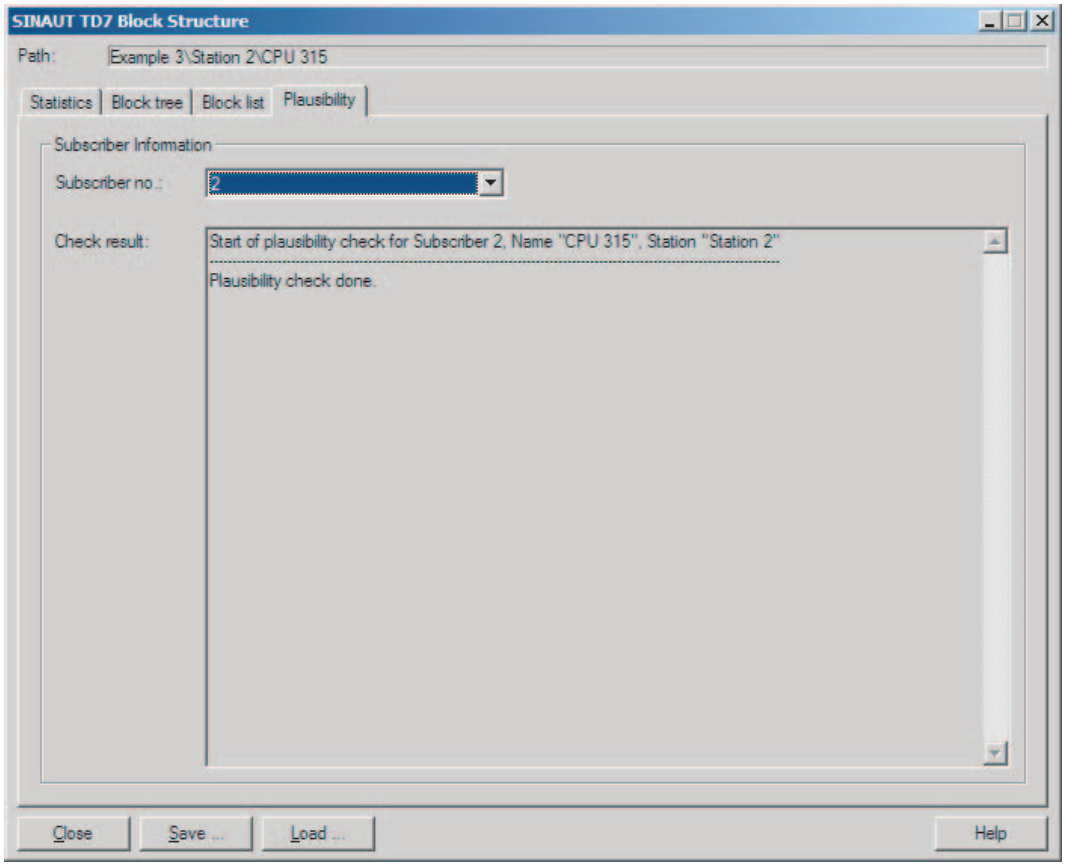


Figure 3-27 *SINAUT TD7 Block Structure* dialog, *Plausibility* tab

3.3.7 TD7 Block Structure for all CPUs

Description of the functions

The *Block Structure crore CPUs* function starts the CPU block structure analysis for the entire project. This function does not access the modules but rather the project data.

It is, for example, possible to compare a CPU of the current project with the CPU of another project by opening the *Block Structure for all CPUs* dialog a second time in the SINAUT Diagnostics and Service tool and copying the data of the other CPU whose block structure data was previously saved as an XML file into the dialog using the *Load* function. The two open dialogs can be placed next to each other (overlapping if necessary) and compared.

The results of the block structure analysis are displayed in four tabs:

- *Statistics* tab:
  This indicates the number, name, type and station of the selected subscriber and displays the communication partner and number of objects of the communication links. The CPU is selected in the tab.
- *Block tree* tab:
  This displays a tree structure with information on the paths and all SINAUT blocks and the data of all SINAUT block calls for the subscribers of the entire project
- *Block list* tab:
  Presents the data of all SINAUT block calls in the form of a list
- *Plausibility* tab:
  Shows the results of plausibility checks for the calls of the SINAUT blocks *BasicTask*, *Startup, Safe* and *ListGenerator* and for the unique assignment of DP numbers, partner numbers and partner object numbers

In all tabs, the block structure of the project can be saved in the form of an XML file using the *Save* function.
The block structure of a different version, a different subscriber or project that was previously saved as an XML file can be displayed in all tabs of the open dialog using the *Load* function. The previous view is then overwritten.

Note

Since the blocks of all CPU modules must be decompiled into STL source files for the *TD7 Block Structure for all CPUs* function, this function can take a considerable time in extensive projects.

Operator activities

1. Open the dialog by clicking on the *SINAUT / TD7 Block Structure for all CPUs* menu.
2. Select the relevant tab to display the information.

Statistics tab

The *Statistics* tab lists the number, name, type and station of the selected subscriber in the *Subscriber Information* area on the left. Select the subscriber in the *Subscriber no.* list box at the top left in the tab.

The *Communication Links* area on the right displays the communication partners of the selected CPU and the number of communication objects per communication partner (target subscriber).

Block tree tab

The *Block tree* tab shows the information obtained on the project path and the path to all subscribers in a tree structure. The following information is listed:

- Project Information (__*Info*)
  with general information on the project path and the logical path
- All subscribers with *Subscriber* number / *Station* name,
  with a listing of the system blocks and user objects freed subscriber with the following information:
  - All SINAUT data blocks (__*Blocks*) in the program directory of the CPU
  - Information (__Info) on the module name, station name, type name, and subnumber of the subscriber
  - The system blocks. These can be: BASICTASK, PARTNERMONITOR, SAFE, PARTNERSTATUS, PULSECOUNTER, STARTUP, TIMETASK, STARTUP
  - The user objects (analog value, binary value, command, counter, setpoint objects etc.) with
    - information on their call data and a
    - listing of the most important parameters of the individual user objects

Block list tab

The *Block list* tab presents the data of the SINAUT block calls of all subscribers in the form of a list. As default, the blocks are displayed in the following five columns:

- **Subscriber**: This contains the block icon and the subscriber number of the relevant CPU. The system blocks have a blue icon, the user blocks a yellow icon.
- **Object name**: Name of the system block or the user object
- **Objectno.**: Object number of the SINAUT object, the number of the instance DB used on the CPU
- **Partnerno.**: Subscriber number of the SINAUT destination or source subscriber as part of the SINAUT addressing (subscriber number, object number)
- **Partnerobjectno.**: Destination or source object number as part of the SINAUT addressing

You can change the ascending order of the blocks within the block list by clicking on the column headers to sort according to the parameters of the individual columns.

Further parameters are added to the view of the block list as follows:

1. Right-click within the tab and click on the *Add columns* context menu.
2. In the *Add columns* dialog that opens, click on the required parameters and confirm with the *OK* button.
   The added block parameters are not included permanently in the list view and are no longer present the next time you open the dialog.

Note

The data of the block list are required for the configuration of a SINAUT ST7cc/sc control center and can be saved for this purpose in an XML file.

Plausibility tab

The *Plausibility* tab runs a series of CPU-specific plausibility checks for a subscriber to the selected and displays the *Check result*. Select the subscriber in the *Subscriber no.* list box at the top in the tab. The following is checked:

- Whether the *Startup* SINAUT block was called correctly,
- Whether the *Safe* SINAUT block needs to be called and was actually called,
- Whether the object numbers (DB numbers) of the SINAUT blocks were assigned uniquely,
- Whether the partner numbers and partner object numbers were assigned uniquely,
- Whether the *BasicTask* SINAUT block was called correctly,
- Whether the *ListGenerator* SINAUT block needs to be called and was actually called.

If errors occur in the plausibility checks listed above, a corresponding message is displayed in the *Check result* text box.

3.3.8 TD7 CPU Program Comparison

Description of the functions

The *TD7 CPU Program Comparison* function displays the results of the comparison of the program of two CPU modules. This function does not, however, access the data of the modules but rather the project data.

When you open the dialog, the CPU selected in the *SINAUT subscriber list* is selected as CPU 1. To make the comparison, select either a further CPU of the same project or any other CPU whose program overview (block structure) was previously saved as an XML file.

The *CPU Program Compare Result* dialog that then opens displays the names and stations of the compared CPU modules in the *Compared Stations / CPUs* area. On the left-hand side of the *Results* area below this, the components and subcomponents (TD7 objects) of the two CPU modules are listed with a symbol indicating the status of the comparison.

Table 3- 5 Significance of the comparison symbols for components in a CPU program comparison

Symbol	Status
	The objects found are different
	The objects found are identical
	The object only serves informal purposes

When you select a component in the list on the left, the parameter names of the components and a comparison symbol are displayed for CPU 1 and CPU 2 in the right-hand part of the *Results* area. If the name of a parameter is identical on both CPU modules, the names displayed only under CPU 1. The comparison symbols have the following meaning:

Table 3- 6 Significance of the comparison symbols for parameters in a CPU program comparison

Symbol	Meaning
	Subcomponents exist only on CPU 1
	Subcomponents exist only on CPU 2
	Components on CPU 1 and 2 are different
	Components are identical on both CPUs

The data of the program comparison can be saved in an XML file. In the same way, data of an earlier program comparison that was saved in an XML file can be displayed again in the dialog using the *Load* button. The previous display is then overwritten.

If you selected the wrong file when loading, an error message is displayed and the dialog display is deleted.

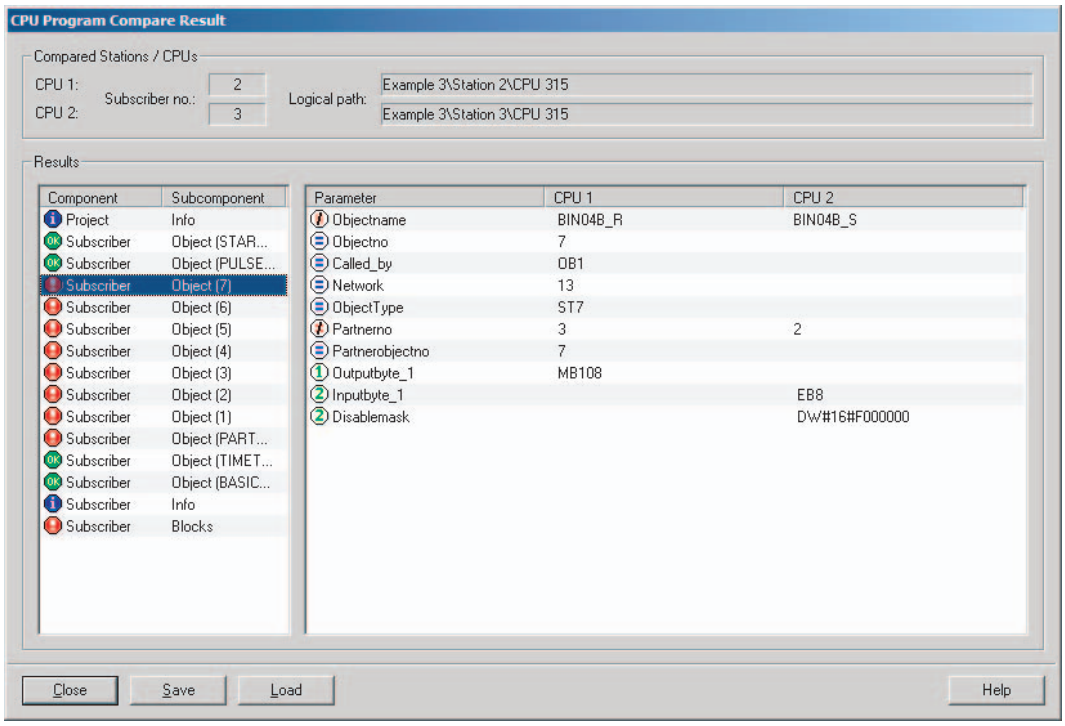


Figure 3-28 *CPU Program Compare Result* dialog

Operator activities

1. Select a CPU in the *SINAUT subscriber list* of the open project. This is identified as CPU 1 in the following dialog.
2. Select the *TD7 CPU Program Comparison* function by selecting the *SINAUT / TD7 CPU Program Comparison* menu. The *Compare TD7 Block Structures* dialog opens.

3. Select a CPU of the project has CPU 2 for comparison in the *CPU 2* box, below the default option *Use CPU*.
4. As an alternative, select *Use file* in the *CPU 2* box and open a previously saved XML file with the block data of any other CPU module by clicking on the *...* button.
5. Click on the *Compare* button. The *CPU Program Compare Result* dialog opens

Note

If you use the XML file with the block structure data of a CPU other than *CPU 2* in the *Compared TD7 Block Structures* dialog, the XML file must only contain the data of **one** CPU. Otherwise, the dialog with the compare results will not open.

3.3.9 TD7 Check of the Communication Configuration

Description of the functions

The *TD7 Check of the communication configuration* functions is used to compare communication structures of two CPU modules.

When you open the dialog, the CPU selected in the *SINAUT subscriber list* is selected as CPU 1. To make the comparison, select either a further CPU of the same project or the program overview (block structure) of any other CPU that was previously saved as an XML file.

In the *Compared Stations / CPUs* area of the *CPU Program Compare Result*, the name and station of the compared CPU modules are displayed.
In the *Results* area below, you can see the objects of the two CPU modules and a symbol indicating their comparisons status into columns on the left-hand side.

Table 3- 7 Comparison symbol for components of the TD7 check of the communication configuration

Symbol	Status
	The objects found are different
	The objects found are identical
	The object only serves informal purposes

When you select a pair of objects in the list on the left, the names of the relevant object parameters and a comparison symbol are displayed for CPU 1 and CPU 2 in the right-hand part of the *Results* area. The comparison symbols have the following meaning:

Table 3- 8 Comparison symbol for parameters of the TD7 check of the communication configuration

Symbol	Meaning
	Subcomponents exist only on CPU 1
	Subcomponents exist only on CPU 2
	Components on CPU 1 and 2 are different
	Components are identical on both CPUs

Since parameters of the two CPU modules with the same functions are listed side by side in the right-hand part of the *Results* area, the objects of the two CPU modules must be crossed over to compare their communication plausibility. For example, the *object* number X of CPU 1 is the *partner object* number X of CPU 2 and vice versa.

The data resulting from comparing the communication configuration of two CPU modules can be saved in an XML file. In the same way, data of an earlier comparison of the communication configuration that was saved in an XML file can be displayed again in the dialog using the *Load* button. The previous display is then overwritten.

If you selected the wrong file when loading, an error message is displayed and the dialog display is deleted.

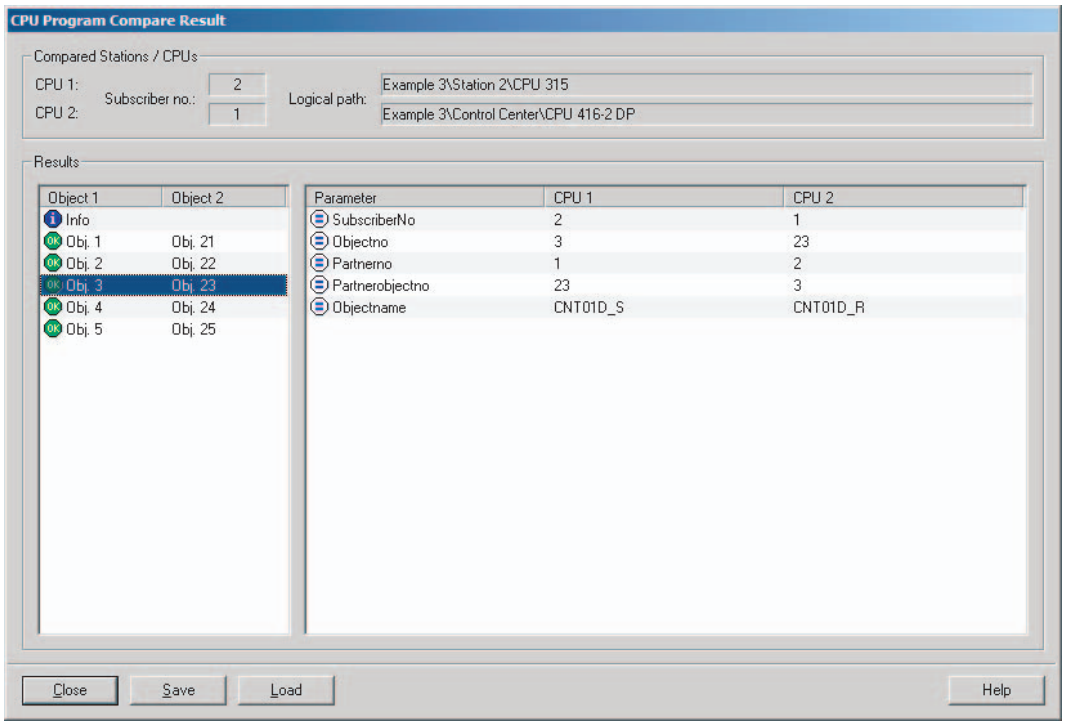


Figure 3-29 *CPU Program Compare Result* dialog

Operator activities

1. Select a CPU in the *SINAUT subscriber list* of the open project. This is identified as CPU 1 in the following dialog.
2. Open the *Compare TD7 Block Structures* dialog by selecting the *SINAUT / TD7 Check of the Communication Configuration* menu.
3. Select a CPU of the project has CPU 2 for comparison in the *CPU 2* box, below the default option *Use CPU*.
4. As an alternative, select *Use file* in the *CPU 2* box and open a previously saved XML file with the block data of any other CPU module by clicking on the *...* button.
5. Click on the *Compare* button. The *CPU Program Compare Result* dialog opens

Note

If you use the XML file with the block structure data of a CPU other than *CPU 2* in the *Compared TD7 Block Structures* dialog, the XML file must only contain the data of **one** CPU. Otherwise, the dialog with the compare results will not open.

3.3.10 TD7onTIM diagnostics

Description of the functions

Note

The *TD7onTIM Diagnostics* function is available only for TD7onTIM-compliant TIM modules on which parameters were set for the *TD7onTIM* software (for example TIM 3V-IE).

TD7onTIM Diagnostics provides information on the status of the data transmission of the *TD7onTIM* software package of the TIM module selected in the SINAUT subscriber list.

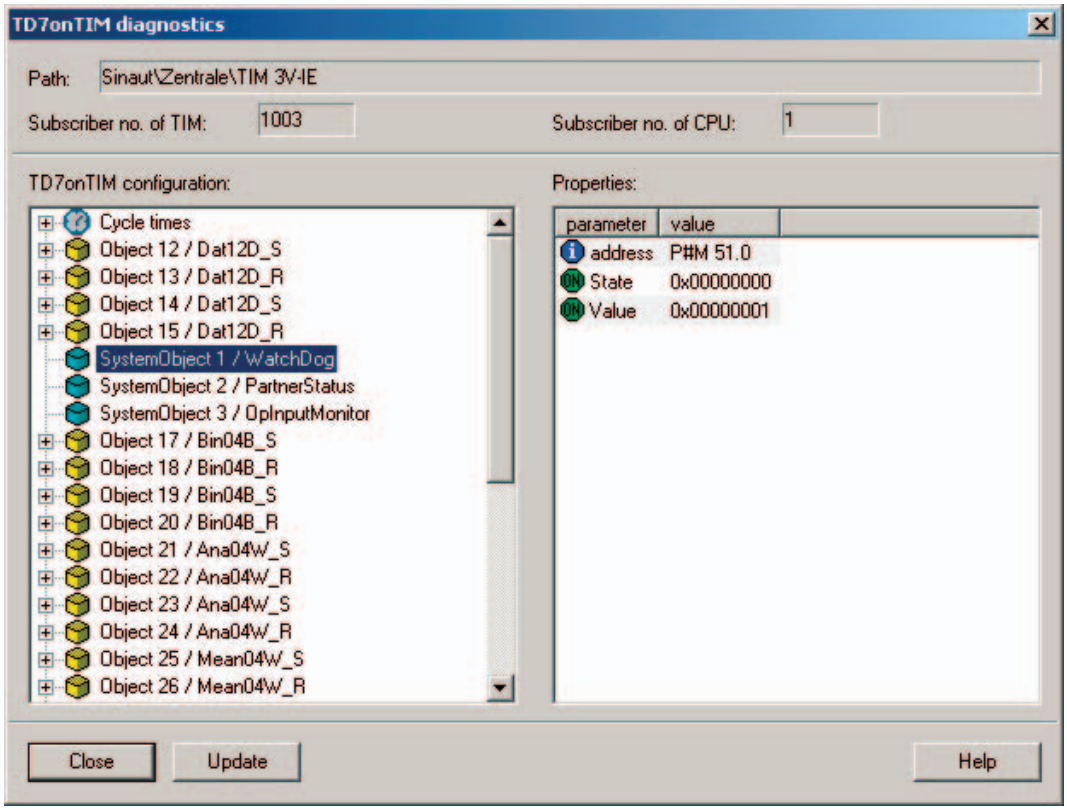


Figure 3-30 *TD7onTIM Diagnostics* dialog with system object numbers based on the example *WatchDog*

In keeping with the parameter assignment of *TD7onTIM*, the diagnostic functions are displayed for the following objects:

- Status of system objects
- Status of data objects
- Status of the input and output channels

The dialog displays the following information for the station of the selected TIM module:

- The *path* of the TIM in the project
- The *subscriber no. of the TIM*
- The *subscriber no. of the CPU*

The lower part of the dialog displays the SINAUT objects and channels with their parameters:

- *TD7onTIM Configuration*:
  This area lists the following directories of the selected TIM as they are successively expanded:
  - System objects (blue symbols)
  - Data objects (yellow symbols)
  - The cycle time
  - Send and receive channels
- *Properties*:
  This area displays the following properties of an object selected in the directory tree:
  - *Parameter name*
  - *Value*
  - *Comment*

The *Parameter name* column lists the individual parameters with colored symbols indicating the following status:
- Blue symbols: Configured data
- Red symbols: Online data

The cycle time is displayed in the *TD7onTIM Configuration* area below the last data object. This is the current time of a sampling cycle in which TD7onTIM samples the work memory of its local CPU. If you click on the cycle time, the corresponding value is displayed in the *Properties* area.

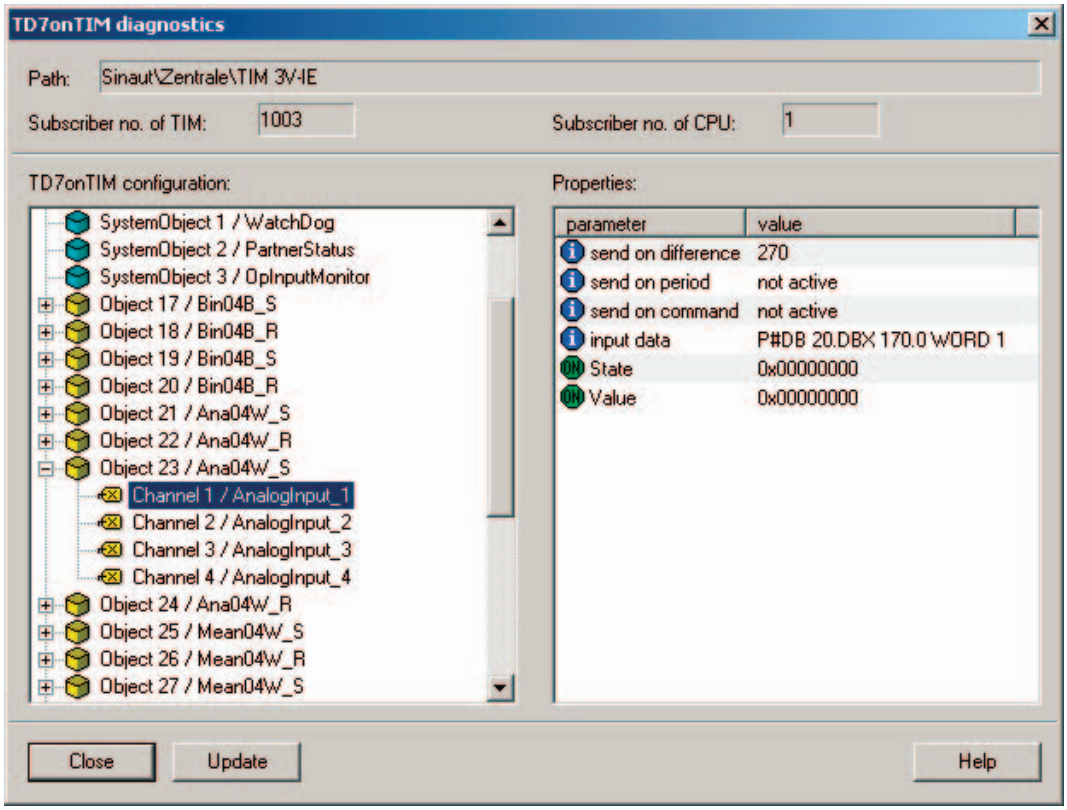


Figure 3-31 *TD7onTIM Diagnostics* dialog with channel parameters based on the example of the *Analog send* channel

Operator activities

1. Select a TD7onTIM-compliant module in the *SINAUT subscriber list* of the open project.
2. Open the dialog by selecting the *SINAUT / TD7onTIM Diagnostics* menu. The dialog opens.
3. Expand the directory tree in the *TD7onTIM Configuration* area.
4. Select a SINAUT object or a send or receive channel in the opened directory tree. The relevant parameters are displayed in the *Properties* area.

3.3.11 SDB Viewer

Description of the functions

The *SDB Viewer* function lists the content of the system data blocks (SDBs) of a previously selected CPU or TIM module.

For TIM modules, the following SDB classes can be selected for display:

- SDB0
- WAN data

- Subscriber data
- Connection data
- LAN connections
- TD7onTIM data (only TIM modules on which parameters have been set for TD7onTIM)
- Ethernet data (only TIM modules that are connected to Ethernet)
- Routing data
- Connection data (PBC)

Apart from SDB0, the representation of the SDB data for the TIM modules is in plain text (as default), you can, however, change to hexadecimal display. The CPU DBs are always displayed in hexadecimal and the corresponding button cannot be deactivated here.

When accessing the subscriber using the *SINAUT subscriber list*, you have the option of displaying the system data blocks from the module (*online*) or from the project (*offline*). There may be differences between online and offline access.

The content of all system data blocks can be saved as a text file.

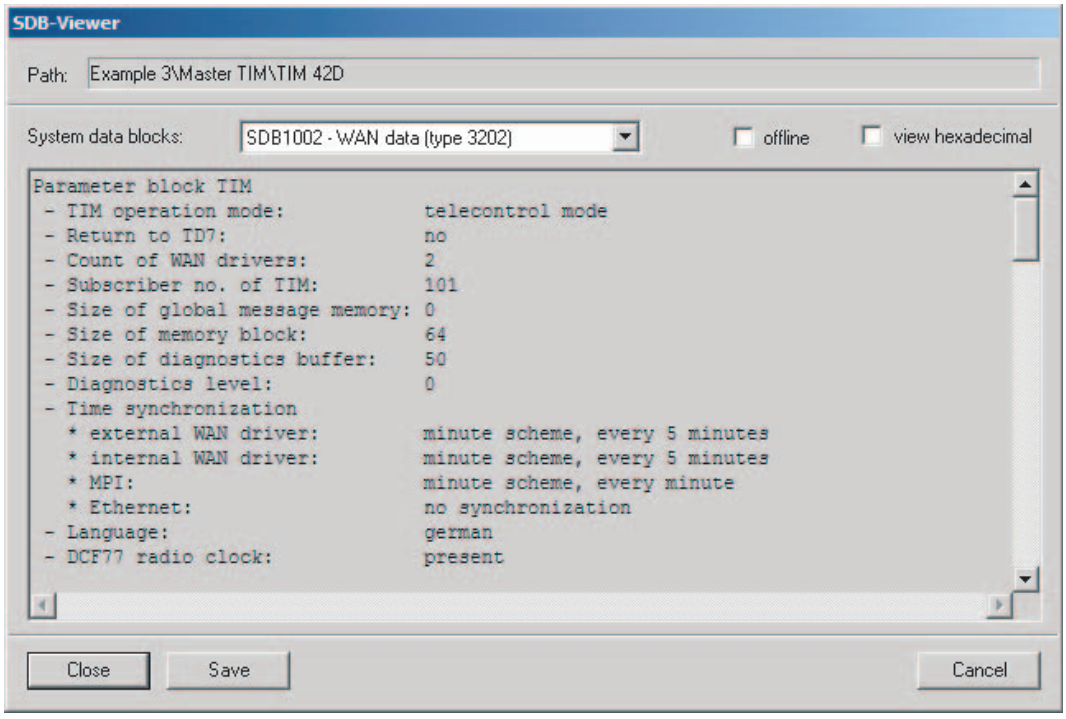

SDB-Viewer
Path: Example 3\Master TIM\TIM 42D
System data blocks: SDB1002 - WAN data (type 3202)
offline
view hexadecimal
Parameter block TIM
- TIM operation mode: telecontrol mode
- Return to TD7: no
- Count of WAN drivers: 2
- Subscriber no. of TIM: 101
- Size of global message memory: 0
- Size of memory block: 64
- Size of diagnostics buffer: 50
- Diagnostics level: 0
- Time synchronization
* external WAN driver: minute scheme, every 5 minutes
* internal WAN driver: minute scheme, every 5 minutes
* MPI: minute scheme, every minute
* Ethernet: no synchronization
- Language: german
- DCF77 radio clock: present
Close
Save
Cancel

Figure 3-32 *SDB Viewer* dialog. In the example, *SDB1000 - WAN data* is selected.

Operator activities

1. Select a subscriber in the *SINAUT subscriber list* of the open project or in *Accessible Nodes*.
2. Open the *SDB Viewer* dialog by selecting the *SINAUT / SDB Viewer* menu.

3. To access data online on the connected module (instead of the project data), deselect the *offline* option on the right to the dialog. The data display is updated immediately with the current module data.
4. Select the required system block class (SDB0, WAN data etc.) in the *System data blocks* list box.
5. Click on the *Save* button to save the content of all system data blocks as a text file.

SDB Viewer - SDB0

SDB0 is displayed in hexadecimal format and contains information on the following points:

- Communication parameters for the MPI bus
- Parameters for the time response of the MPI bus
- MPI address of the module
- Rack configuration with rack and slot addresses for S7-300 stations

WAN data SDB

The WAN data SDB contains information on the parameter assignment of the TIM module and the WAN drivers as illustrated below based on an example.

Table 3- 9 Example of the information of a WAN data SDB (type 3202)

Parameter block	Parameter
Parameter block TIM	- TIM operation mode: telecontrol mode - Return to TD7: no - Count of WAN drivers: 2 - Subscriber no. of TIM: 1002 - Size of global message memory: 0 - Size of memory block: 64 - Size of diagnostics buffer: 50 - Diagnostics level: 0 - Time synchronization * external WAN driver: minute scheme, every 5 minutes * internal WAN driver: minute scheme, every 5 minutes * MPI: minute scheme, every minute * TIM bus: no synchronization - Language: german - DCF77 radio clock: not present - Size of ram disk: 0 - Minimum heap reserve: 85 - TIM bus present: 0 - No of master: 1
Parameter block WAN driver 1	Base parameters: - Interface: external - TIM type: station - Net type: dial-up network - Operating mode: spontaneous - Message format: FT1.2 - Acknowledgment: short acknowl. - Retry factor: 7 - WAN protocol: ST7

Parameter block	Parameter
	- Master/node station/station no.: 2 - Max. message length: 240 - Number of local CPUs: 1 - General request priority: 0 - Number of spontaneous messages: 0 - Baud rate: 38,400 - Call answer delay: 0 - Country mode: Germany - Dialing mode: AT mode - Dialing format: 8 data bits, no parity, 1 stop bit - Extra transmission time: 0 - Customer identification: 0 - Paramet. for cond. spont. mess.: standard conditions - Limit for locked messages: 0 - Transfer mode: send single messages - Cancel delay time: 0 - Operating mode: Interrupt (Block) - # of subscribers: 2 Telephone number list: - Block 1 own telephone number * Telephone number 1 Telephone number: 2 - Block 2 remote telephone numbers Redialing attempts: 3 Cancel parameter: 0 * Telephone number 1 driver type master No special service Station address: 1 Telephone number: ATDP3 AT string: ATS45=3\N0F0&W
Parameter block WAN driver 2	Base parameters: - Interface: internal - TIM type: station - Net type: dedicated line - Operating mode: polling - Message format: FT1.2 - Acknowledgment: short acknowl. - Retry factor: 3 - WAN protocol: ST7 - Master/node station/station no.: 2 - Max. message length: 240 - Number of local CPUs: 1 - General request priority: 0 - Number of spontaneous messages: 20 - Number of permanent messages: 0 - Station address of cyclic partner: 1 - Baud rate: 19,200 - Polling time: 0 - RTS/CTS delay time: 0 - Send delay time: 0 - Extra transmission time: 0 - Limit for locked messages: 0

Parameter block	Parameter
	- Operating mode: Interrupt (Block) - # of subscribers: 2

Subscriber data SDB

The subscriber data SDB contains information on the settings for data of all subscribers of a project. The content of a subscriber SDB is shown below based on an example.

Table 3- 10 Example of the information of a subscriber data SDB (type 3203)

Subscriber	Parameter
# of subscribers:	5
Subscriber 1	Subscriberno.: 1 Subscriber info: 0x0 ST7-CPU additional info: 0x0 send NO status object to subscriber count of connection blocks: 2 device | subnet ID | CFB/MPI | state | connection | STA int. WAN | 008a00000001 | 0 | remote | WAN connection.,ST7 | 0x01 ext. WAN | 008a00000001 | 0 | remote | WAN connection.,ST7 | 0x01 count of partner blocks: 1 subscriber no.: 2
Subscriber 2	Subscriberno.: 1003 Subscriber info: 0x1 ST7 TIM etc.

Connection data SDB

The connection data SDB contains information on the parameter settings of the local X connections of a TIM module to their CPU. This is illustrated below based on an example.

Table 3- 11 Example of the information of a connection data SDB (type 3205)

count of X com. blocks:	1		
block	connection type	loc. device ID	target MPI
1	static	0	2

LAN connection SDB

The LAN connection SDB contains summarized information on the parameter assignment of all LAN connections on a TIM module. This is illustrated below based on an example.

Table 3- 12 Example of the information of a LAN connection SDB (type 3201)

local subscriberno.:	1002
local MPI address:	3
count of LAN blocks:	2

```
                                block   connection time     CFB/MPI  size of queue
                                1       X com not conf.     2        64
                                2       PBC                 4        64
count of TIM blocks:            1
                                Sno.    MPI   additional info
                                1001    8     0x0
DCF77 NOT built in
CPU slave
external WAN driver NOT active
internal WAN driver NOT active
count of local partners:        2
                                MPI     Rack  Slot
                                2       0     2
                                3       0     7
```

TD7onTIM data SDB

The TD7onTIM data SDB contains information on the parameter assignment of the SINAUT objects, their send and receive channels and the source or destination subscribers. An excerpt of this information is shown below based on an example.

Table 3- 13 Example of the information of a TD7onTIM data SDB (type 3206)

```
Parametertype T4T SDB Main Header = 61185
Length of block:                                12
Count of target subscribers:                    3
Count of source subscribers:                    1
Count of objects:                               7
Count of system objects:                        3
-------------------------------------------------
----------------------
Parametertype T4T SDB Target Subscriber = 61186
Length of block:                                12
Type of target subscriber:                      4
Subscriberno.:                                  8
Supervision time:                               900
Timestamp:                                      1
-------------------------------------------------
----------------------
Parametertype T4T SDB Target Subscriber = 61186
Length of block:                                12
Type of target subscriber:                      0
Subscriberno.:                                  2
Supervision time:                               900
Timestamp:                                      1
-------------------------------------------------
----------------------
Parametertype T4T SDB Source Subscriber = 61187
Length of block:                                30
Subscriberno. of TIM:                           1003
MPI address of TIM:                             8
Subscriberno. of CPU:                           4
```

```
MPI address of CPU:                                    7
Ext. Time Stamp:                                       0
Max Spon Out:                                          3
Max Main In:                                           2
Max Sub In:                                            5000
Scan Delay:                                            100
Max Input Time:                                        0
Input Delay Time:                                      0
Max Connect Time:                                      0
Address Check:                                         1
-------------------------------------------------------
----------------------
Parametertype T4T SDB System Object = 61188
Length of block: 12
Type of system object: 32,512
DBNo | memLoc | byteAdr | bitAdr|
 7 | DB | 1 | 0 |
-------------------------------------------------------
----------------------
Parametertype T4T SDB System Object = 61188
Length of block: 28
Type of system object: 32,513
DBNo | memLoc | byteAdr | bitAdr|
 7 | DB | 2 | 0 |
Partner | Subscriberno.
 1 | 8
 2 | 2
 3 | 0
 4 | 0
 5 | 0
 6 | 0
 7 | 0
 8 | 0
-------------------------------------------------------
----------------------
Parametertype T4T SDB Partner = 61190
Length of block: 6
Partnerno.: 1
-------------------------------------------------------
----------------------
Parametertype T4T SDB Partner = 61190
Length of block: 6
Partnerno.: 8
-------------------------------------------------------
----------------------
Parametertype T4T SDB Channel = 61191
Length of block: 48
Channel active: 1
Channel type = Message Send = 60929
Send On Difference: 1
Send On Period Active: 0
Send On Period: 10
Send On Command Active: 0
```

```
Send On Command:
DBNo | memLoc | byteAdr | bitAdr|
 0 |UNKNOWN| 0 | 0 |
Alarm Mask: 0
Send Buffer Principle Mask: 0
Disable Mask: 0
Input Data:
dataType | repeatF | dbNo | memLoc | byteAdr |
bitAdr|
 BYTE | 1 | 0 | M | 11 | 0 |
-----------------------------------------------------
----------------------
```

Ethernet data SDB

The Ethernet data SDB contains information on the IP address of the selected TIM module, the subnet mask and any configured router. If, as in the case here, the address of the router is the same as the IP address of the TIM, no router is set. The IP addresses are set in network configuration.

Table 3- 14 Example of the information of an Ethernet data SDB (type 3100)

```
IP address       : 140. 80. 0. 3
subnet mask      : 255. 255. 0. 0
default router   : 140. 80. 0. 3
```

Routing data SDB

The routing data SDB contains information on the individual subnets of a project. This is illustrated below based on an example.

Table 3- 15 Example of the information of a routing data SDB (type 3002)

Subnet type	**ID, address**
local subnet	- local subnet ID : 47 11 00 00 00 0c - local device ID : 03
local subnet	- local subnet ID : 47 11 00 00 00 14 - local device ID : 02
remote subnet	- remote subnet ID : 00 8a 00 00 00 01 - next station addr : 01 - local device ID : 03
etc.	

Connection data (PBC) SDB

The connection data (PBC) SDB contains information on the PBC connections of a TIM module. This is illustrated below based on an example.

Table 3- 16 Example of the information of a connection data (PBC) SDB (type 700)

```
STEP 7 connection   - Connection setup: fixed configured, static
```

```
- Connection type: active connection setup
- Operating Mode: send no operating mode messages
- Connection ID: 1
- local device ID : 01
- local TSAP ID: 11 04
- remote Station addr.: 02
- remote TSAP ID: 11 04
```

Consistent data SDB

The consistent data SDB is used to check consistency of the SDBs. Based on these SDBs, the TIM can check the consistency of the SDBs generated for it.

Table 3- 17 Example of the information of a consistency SDB (type 3118)

SDB no.	Type	Time stamp
1000	2	07/14/05 17:11:11
1001	3	07/14/05 17:11:11
1002	5	07/14/05 17:11:11
1003	1	07/14/05 17:11:11
1004	6	07/14/05 17:11:11
1005	7	07/14/05 17:11:11
1006	8	07/14/05 17:11:11
1008	1024	07/14/05 17:11:11

3.4 Service functions

3.4.1 Download SDB

Description of the functions

The *Download SDB* function downloads the system data blocks of a TIM module from the program directory of the SIMATIC Manager to the module. This is the same procedure as the *Download Module* function in the SIMATIC Manager.

To activate newly downloaded SDBs on a TIM module, the TIM must be restarted.

Note

When a TIM module is restarted after downloading new SDBs, the connection between the TIM and other SINAUT partners (SINAUT connections) is terminated. This leads to error messages on the partners of the TIM module. In the case of a node TIM, the connections to the downstream stations are also reported as being disrupted.

With a node TIM, any data messages stored on the TIM are lost during the restart. This can be important, particularly in dial-up networks.

When downloading SDBs to TIM modules, you should therefore note the following points:

- Before you transfer the SDBs, you should give the TIM the opportunity of transferring any messages stored on it.
- After restarting the TIM, the SINAUT connections are established again automatically, the connection between the PG and the TIM must, however, be activated by the user on the PG if it is required.

More detailed information refer to the section *Installing and putting a TIM into operation* / *Configuring and assigning parameters for a TIM*.

Operator activities

1. Select a TIM module in the *SINAUT subscriber list* of the open project.
2. Start the function by selecting the *SINAUT / Download SDB* menu. The *Open* dialog is displayed.
   Follow the instructions in the subsequent dialogs. When necessary, you can cancel the procedure in these dialogs.
3. After downloading SDBs, the *Open* message dialog asks you when you want to start the module again. To activate the downloaded SDBs, you must restart the module.
4. Click on *Yes* to restart module. A message is displayed indicating that the SDBs were successfully loaded.
5. Confirm this message by clicking on *OK*.

3.4.2 Firmware update

Description of the functions

The *Firmware Update* function allows you to load a new firmware version on a TIM module. The function is supported on TIM modules that have the RMOS for TIM-ST7 operating system as of version 2.04.

Note

You can read out the version of the operating system of a TIM using the *Module Information* function / *General* tab.

To use this function, the firmware must have been installed on the computer using the setup. If the firmware is not installed on the PG or is incomplete, a message is displayed.

By clicking the *Update details* button in the *Firmware Update* dialog, you open the *Update details* dialog that displays the firmware version installed on the TIM module and located on your computer.

After the download, the module is automatically reset to activate the new firmware.

The parameter assignment of the module is not affected by the firmware update.

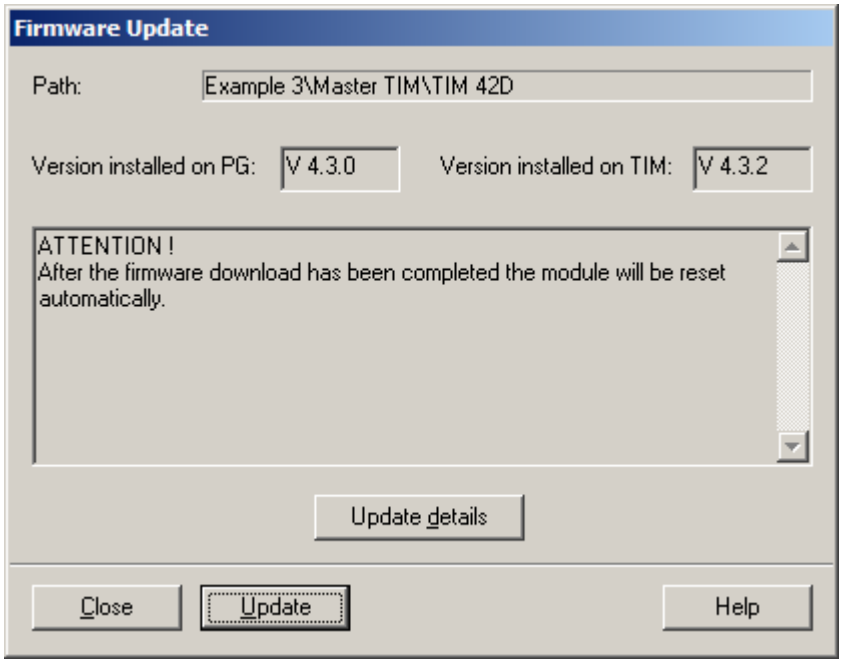


Figure 3-33 *Firmware Update* dialog

Note

Downloading the firmware to the module can take several minutes.

Operator activities

Note

Make sure that you select the correct TIM module in your project. The station and module name must match and the configured module must be of the same type as the module to which you are downloading.
If this is not the case, a dialog will inform you of this at regular intervals during the update. The display of this message interrupts the update until the dialog is acknowledged by clicking on *OK*.

1. Select the relevant TIM in the *SINAUT subscriber list* of the open project.
2. Start the function by selecting the *SINAUT / Firmware Update* menu. The *Firmware Update* dialog opens.
3. Click on the *Update details* button if you require a detailed information on the firmware update. The *Update details* dialog opens.
4. Click on the *Update* button in the *Firmware Update* dialog to start the firmware update. The following dialog *Loading* informs you of the current progress of the update.
5. On completion of the firmware update, a dialog appears with a message to this effect. Confirm the message with *OK*.

Update details dialog

The *Update details* dialog that can be opened from the *Firmware Update* dialog displays detailed information on the firmware update. The function and version of each firmware component on the TIM module and on the PG is shown and you can see whether the relevant component is copied, replaced, ignored, or deleted during the firmware update.

The various actions have the following significance:

- **copy**: The file is copied from the PG to the TIM module.
- **replace**: The file on the TIM module is replaced by the file on the PG.
- **ignore**: The file is not affected by the firmware update.
- **delete**: There is no newer version for the file. The existing file is no longer required and is deleted during the firmware update.

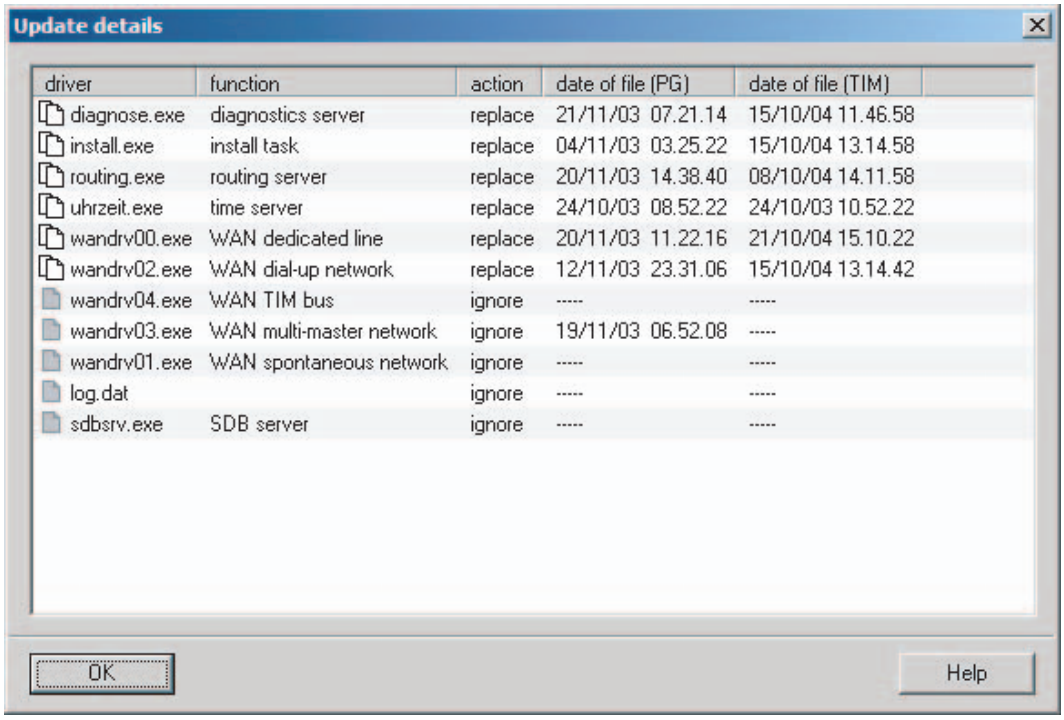


Figure 3-34 *Update details* dialog of the *Firmware Update* function

3.4.3 Repair

Description of the functions

The *Repair* function allows you to restore TIM modules with a defective flash disk. The *Firmware Update* function is available for loading firmware on a functioning TIM module.

The *Repair* function runs a completely new installation of the firmware on a TIM module.

Note

The *Repair* function should not be used without consulting the offline.

The *Repair* function is supported on TIM modules that have the RMOS for TIM ST7 operating system as of version 2.04.

Note

You can read out the version of the operating system of a TIM using the *Module Information* function / *General* tab.

To use this function, the firmware must have been installed on the computer using the setup. If the firmware is not installed on the PG or is incomplete, a message is displayed.

By clicking the *Installation details* button, you open the *Update details* dialog that displays the firmware version installed on the TIM module and the version on your computer.

Following the download, the module is automatically reset to activate the new firmware.

Steps in repairing

Repairing involves the following steps:

- The flash disk of the TIM module is formatted. After formatting, the TIM module runs a reset.
- This is followed by a default startup.
- Once the wait time for the default startup has elapsed, the firmware version installed on the computer is downloaded to the TIM.
- Following this, the system data blocks are transferred to the TIM module.
- The module is then reset and resumes operation with its full functionality.

Note

While the firmware is being reinstalled, the TIM module has MPI address 3.

Make sure that this address is free on the MPI bus to which the TIM module is connected.

Operator activities

Note

Make sure that you select the correct TIM module in your project. The station and module name must match and the configured module must be of the same type as the module to which you are downloading.
If this is not the case, a dialog will inform you of this at regular intervals during the update. The display of this message interrupts the update until the dialog is acknowledged by clicking on *OK*.

1. Make sure that MPI address 3 is either free or is occupied by the module on which you want to install.
2. Select the TIM you want to repair in the *SINAUT subscriber list* of the open project.
3. Start the function by selecting the *SINAUT / Repair* menu and then *Complete reinstallation* in the context menu. The *Firmware Install* dialog opens. The version installed on the PG is displayed. If the firmware on the PG is incomplete, a message to this effect is also displayed.
4. Click on the *Update details* button if you require a detailed information on the firmware update. The *Installation details* dialog opens.
5. Click on the *Start Installation* button in the *Firmware Install* dialog to start the repair. The following dialog *Download* informs you of the current progress of the procedure.
6. On completion of the repair, a dialog opens with a message to this effect. Confirm the message with *OK*.

Installation details dialog

The *Installation details* dialog that can be opened from the *Firmware Install* dialog displays detailed information on the repair. For each firmware component, the function and version on the TIM module and on the PG are shown and you can also see whether the component will be copied, replaced, ignored or deleted during the repair.

The various actions have the following significance:

- **copy**: The file is copied from the PG to the TIM module.
- **replace**: The file on the TIM module is replaced by the file on the PG.
- **ignore**: The file is not affected by the repair.
- **delete**: There is no newer version for the file. The existing file is no longer required and is deleted during the repair.

3.5 Message protocol diagnostics

Introduction

To read out messages, SINAUT ST7 provides you with the option of recording messages transferred in the CPU, TIM and ST7cc/ST7sc PC components; in other words, to archive them in protocols.

The following protocol types are distinguished:

- Testcopy DB:
  This is used to record messages on a CPU module.
- ST7cc/ST7sc protocol:
  This is used to record messages in SINAUT ST7cc or SINAUT ST7sc.
- TIM message protocol:
  This is used to record messages received and sent by a TIM module.

The message protocols are displayed in a message list in the form of a table.

Note

The functions of *message protocol diagnostics* are used only to analyze message protocols that have already been saved.
Message recording is activated

- for the Testcopy DB by setting up DB99 in the SIMATIC Manager
- for the SINAUT ST7cc/sc protocols by setting up the Textcopy block
- for the TIM message protocol in the *TIM Message Monitor* function of the SINAUT Diagnostics and Service tool

3.5.1 Testcopy DB

Description of the functions

The recording of the message traffic on the CPU is made possible by the TestCopyData data block. With the aid of the FC Testcopy function, you can filter out certain message types that can be copied from the send or receive buffer of the CPU for further evaluation in a TestCopyData data block. The default for the TestCopyData data block is DB99.

When setting the the data block, in the control field of the TestCopyData DB, you set filters for certain message types, subscribers and objects and make further settings. All send and receive messages are stored in the TestCopyData DB in chronological order. For a detailed description of the functions and setting up the TestCopyData DB, refer to the chapter *SINAUT TD7 software package* of the SINAUT ST7 system manual.

With the aid of the *Testcopy DB* function of the SINAUT Diagnostics and Service tool, a TestCopyData data block is opened and the recorded messages displayed as a message list for further analysis.

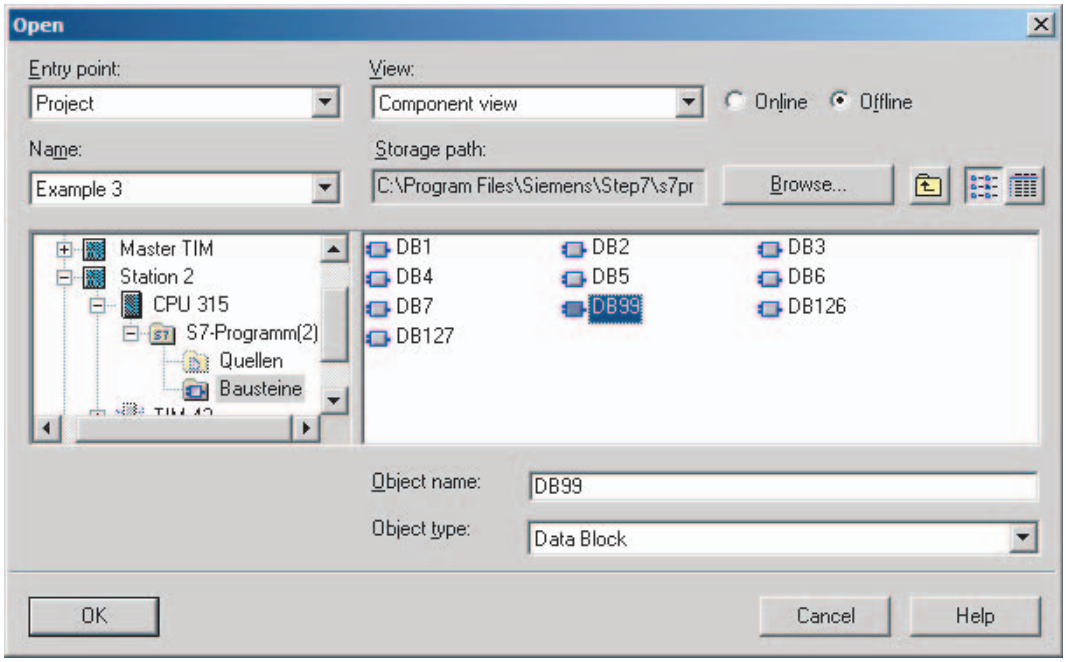


Figure 3-35 *Open* dialog of the *Testcopy DB* function

Operator activities

1. Open the *Open* dialog by selecting the *Project / Testcopy DB* menu.
2. In the *Open* dialog, select the *Online* option at the top right if you want to access the TestCopyData DB directly on the CPU module. This assumes that there is a functioning connection between the PG and the relevant CPU module.
3. In the *Entry point* list box, specify *Project* or *Accessible nodes* as the project type.
4. Select the required project in the *Name* list box or click on the *Browse* button if the project is not available in the *Name* list box.
   The *Browse* dialog opens with the file tree of the PG/PC. Enter the directory path in the *Search in directory* input box and then click the *Start search* button. Select the project you require in the *User Projects* tab on the right and click on the *OK* button. You return to the *Open* dialog.
5. On the left, in the folder list open the list of project stations by double-clicking on the project name and then double-click to select the following:
   - the required station
   - its CPU module
   - the S7 program and
   - the block folder.

   The individual objects of the selected block folder are displayed in the object list on the right. As default, only data blocks are displayed, since the object type *data block* is the default in the *Object type* list box below.
6. On the right in the object list, select the DB TestCopyData (DB99), this is entered in the *Object name* input box below.
7. Confirm with *OK* and the message protocol *Testcopy DB* opens. if the Testcopy DB does not include any messages, a message to this effect is displayed.

Working in the open *Testcopy-DB* message protocol is the same as in the TIM message protocol.

3.5.2 ST7cc / ST7sc protocol

Description of the functions

The recording of the message traffic between SINAUT subscribers and an ST7cc or ST7sc control center is possible in ST7cc or ST7sc using the trace. The function of the trace is explained in the *SINAUT ST7cc Control Center Manual*. This contains precise information on starting the trace functions, displaying messages in the output window, activating the trace output files and other functionalities.

Operator activities

1. Open the ST7cc or ST7sc protocol by selecting the *Project / ST7cc/ST7sc Protocol* menu. The *Open* dialog is displayed.
2. In the file tree, select the directory and the *7DS* file of the ST7cc or ST7sc message protocol you require and confirm with the *Open* button.
   The message protocol opens in a separate window.

Working in the open ST7cc or ST7sc message protocol is the same as in the TIM message protocol.

3.5.3 TIM message protocol

Description of the functions

The recording of the TIM messages is started with the *TIM Message Monitor* of the SINAUT Diagnostics and Service tool. The messages received and sent by the routing server of the TIM are recorded.

The *TIM message protocol* function is used only to open a TIM message protocol for subsequent evaluation.

Operator activities

1. Open the TIM message protocol by selecting the *Project / TIM Message Protocol* menu. The *Open* dialog is displayed.
2. In the file tree, select the directory and the *7DT* file of the required TIM message protocol and confirm with the *Open* button. The TIM message protocol is opened in a separate window.

Note

You activate and deactivate recording of the TIM message protocol and specify the name and storage location of the message protocol file using the *TIM Message Monitor* function of the Diagnostics and Service tool.

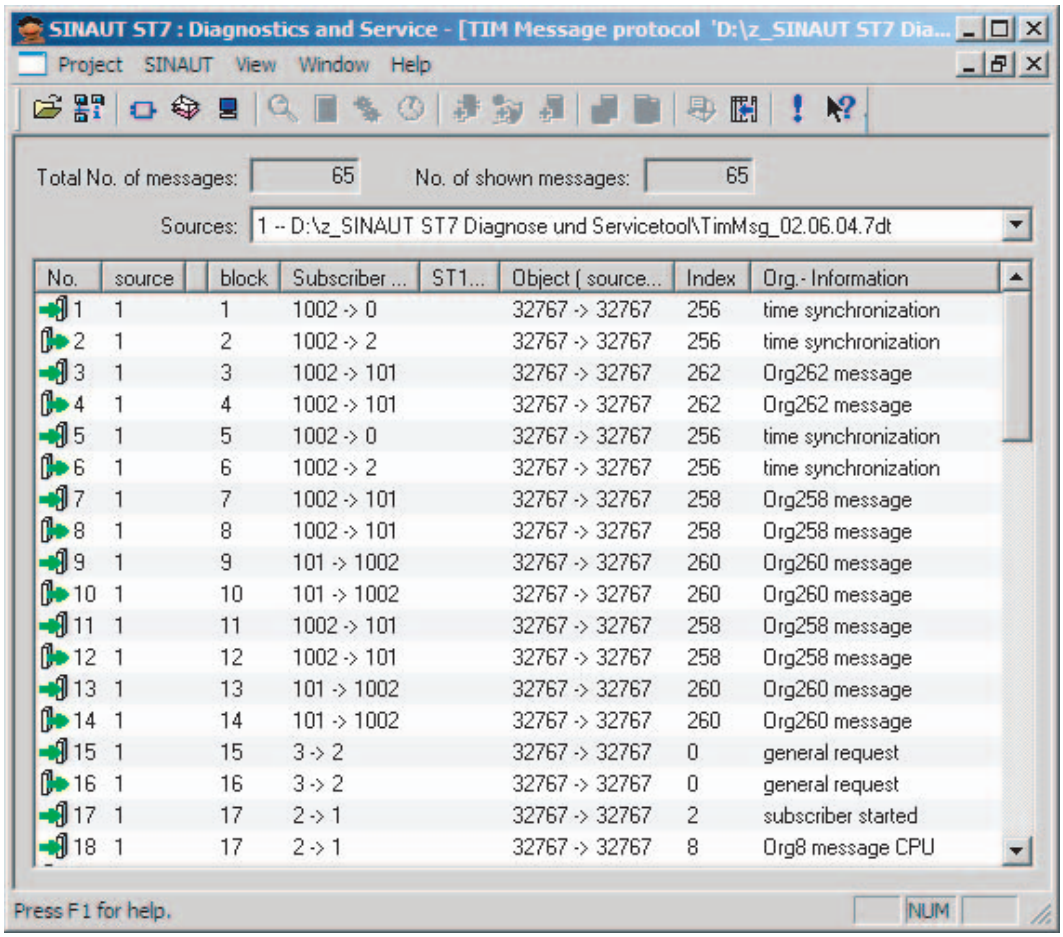
3.5.4 Diagnostics of the TIM message protocol

With a few exceptions relating to format and the diagnostic data, the functions and handling of the TIM message protocol are largely identical with those of the other message protocol types *Testcopy DB* and *ST7cc/ST7sc protocol*. The description can therefore also be applied to the other message protocol types.

Structure of the TIM message protocol

The upper part of the *TIM message protocol* dialog summarizes the following information:
- Total number of messages,
- Number of messages shown,
- Source and path of the protocol file

Below this, there is the **list of TIM messages**, that has nine columns as default and provides the following information on every message:
- A symbol for incoming and outgoing messages
- Message number
- Source: number of the open message protocols in ascending order
- msec: Recording time with DB TestCopyData and ST7cc/ST7sc protocols
- Block: Number of the message block with TIM message protocols
- Subscriber number of the message source and destination
- ST1 message number (only for SINAUT ST1 messages)
- Object number of message source and destination
- Index no.: Address parameters for net data in data messages
- Org. Information


SINAUT ST7 : Diagnostics and Service - [TIM Message protocol 'D:\z_SINAUT ST7 Dia...
Project SINAUT View Window Help
Total No. of messages: 65
No. of shown messages: 65
Sources: 1 -- D:\z_SINAUT ST7 Diagnose und Servicetool\TimMsg_02.06.04.7dt
Press F1 for help.
NUM

No.	source	block	Subscriber ...	ST1...	Object (source...	Index	Org.- Information
1	1	1	1002 -> 0		32767 -> 32767	256	time synchronization
2	1	2	1002 -> 2		32767 -> 32767	256	time synchronization
3	1	3	1002 -> 101		32767 -> 32767	262	Org262 message
4	1	4	1002 -> 101		32767 -> 32767	262	Org262 message
5	1	5	1002 -> 0		32767 -> 32767	256	time synchronization
6	1	6	1002 -> 2		32767 -> 32767	256	time synchronization
7	1	7	1002 -> 101		32767 -> 32767	258	Org258 message
8	1	8	1002 -> 101		32767 -> 32767	258	Org258 message
9	1	9	101 -> 1002		32767 -> 32767	260	Org260 message
10	1	10	101 -> 1002		32767 -> 32767	260	Org260 message
11	1	11	1002 -> 101		32767 -> 32767	258	Org258 message
12	1	12	1002 -> 101		32767 -> 32767	258	Org258 message
13	1	13	101 -> 1002		32767 -> 32767	260	Org260 message
14	1	14	101 -> 1002		32767 -> 32767	260	Org260 message
15	1	15	3 -> 2		32767 -> 32767	0	general request
16	1	16	3 -> 2		32767 -> 32767	0	general request
17	1	17	2 -> 1		32767 -> 32767	2	subscriber started
18	1	17	2 -> 1		32767 -> 32767	8	Org8 message CPU

Figure 3-36 The *TIM Message Protocol* dialog

Functions of the TIM message protocol

Further functions are available with the right mouse button in a context menu. These can be grouped as follows:

- Presentation of the messages
- Details (of the message content)
- Statistics
- Filter functions
- Exporting protocol files

Working with the message list

After opening the *TIM message protocol* with the *Project / TIM Message Protocol* menu, the following options are available in the open dialog:

1. If you click on the header of any column, the message list will be sorted according to this criterion instead of the consecutive number.
2. Right-click (cursor within the protocol window) to activate further functions. A context menu opens with other functions.
3. Select the required function with the left mouse button in the context menu. Each function opens a dialog.

Presentation of the messages

Add new columns

With this function, you can add extra columns in the TIM message list with further message information.

After starting the *Add New Columns* function, the *Add Column* dialog opens in which you can select the properties with the mouse. The selected message properties are displayed as additional columns in the message list.

Delete additional columns

This function deletes all previously added columns. No further dialog is displayed.

Details

The *Details* function provides you with detailed information on the content of the individual messages.

To open the *Details* dialog, you must first select a message. As an alternative to using the right mouse button, the *Details* dialog can also be opened by double-clicking on a message in the list.

In the upper part of the *Details* dialog, you can see the path of the protocol file and five tabs containing further information.

To page to other messages within the *Details* dialog, click on the *<<* or *>>* button. In each tab, the dialog view switches to the previous or next message.

- The *Message Header* tab displays a table containing three columns with the following data from the message header of the selected message:
  - Variable name or short name of the message
  - Value of the individual variables
  - Variable name

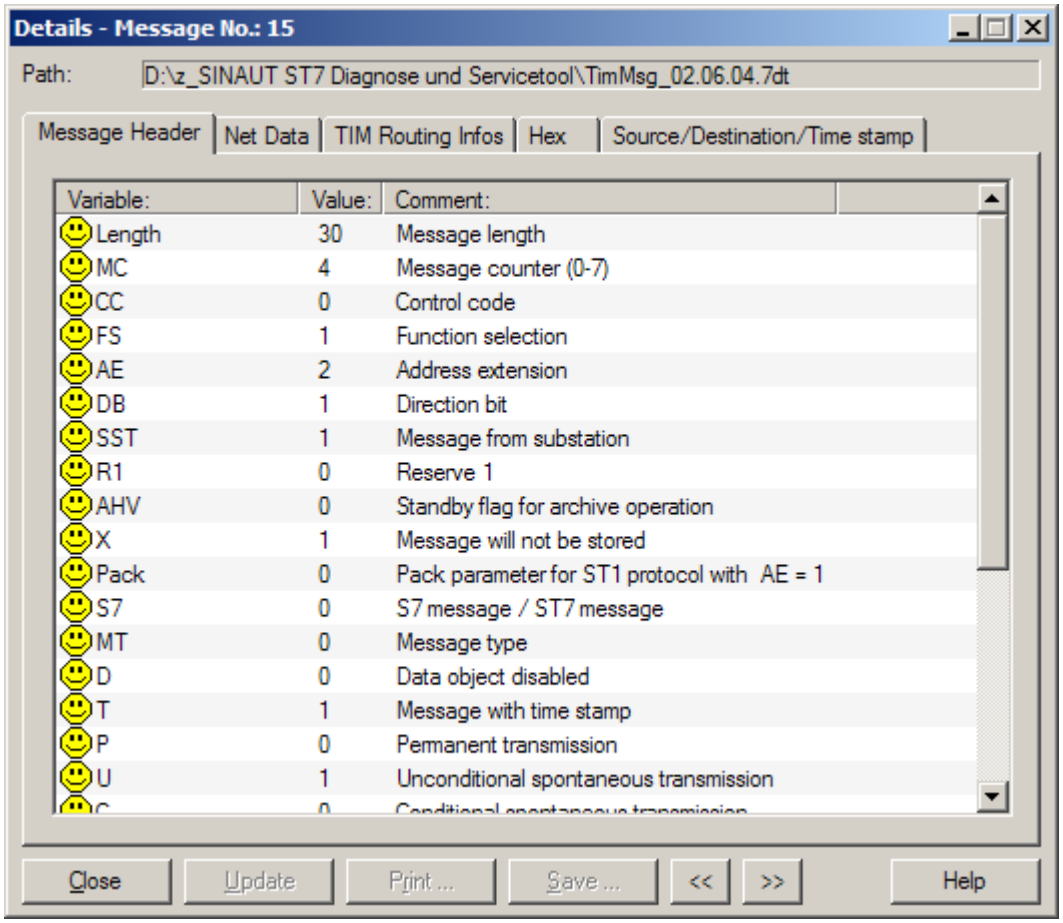


Figure 3-37 *Details* dialog, *Message Header* tab

- The *Net Data* tab shows the net data of the message.
  With the message type 0 and 1 (organizational messages), the data is displayed as plain text.
  with message type 2 and 3 (data messages), the values are displayed. The mode of the display can be changed with the *Display mode* list box using the mouse. The options are:
  - ST7 analog value
  - ST1 analog value
  - ST7/ST1 counted value
  - ST7/ST1 message

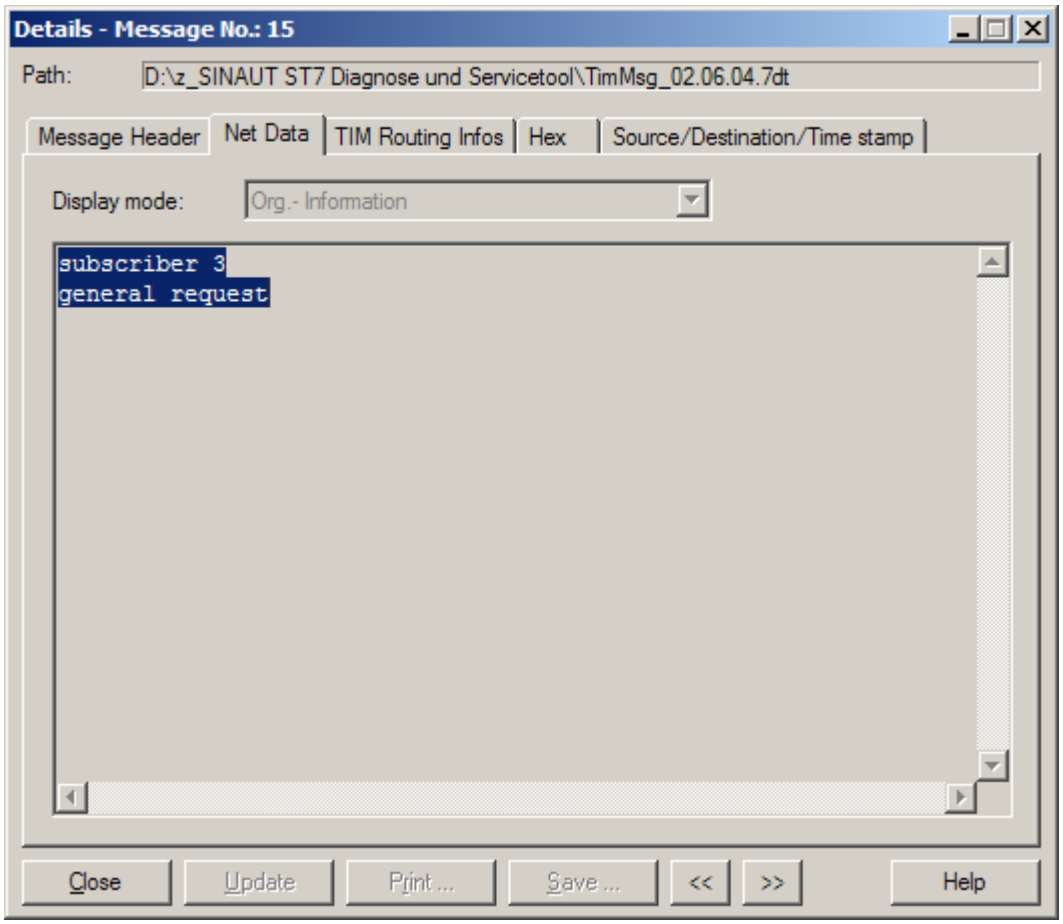


Figure 3-38 *Details* dialog, *Net Data* tab

- The *TIM Routing Infos* tab shows the following:
  - At the top in the *Message* area:
    - the internal task ID
    - the complete message length [bytes]
  - Below in the *Address infos* area:
    - the device ID as a number and in plain text (for example MPI bus)
    - the CN ID
    - the station address

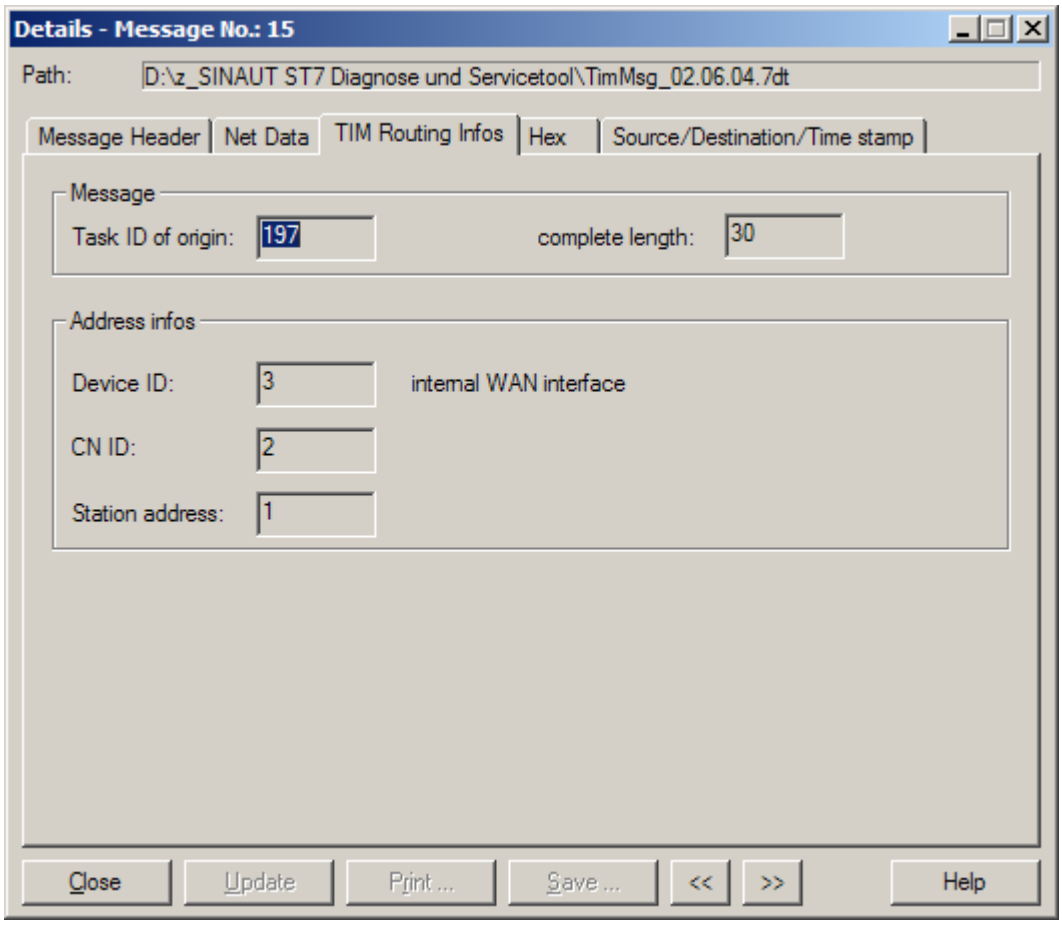


Figure 3-39 *Details* dialog, *TIM Routing Infos* tab

- The *Hex* tab shows the following in hexadecimal format:
  - In the *complete buffer* area, the content of the entire message
  - In the *net data* field, only the net data of the message

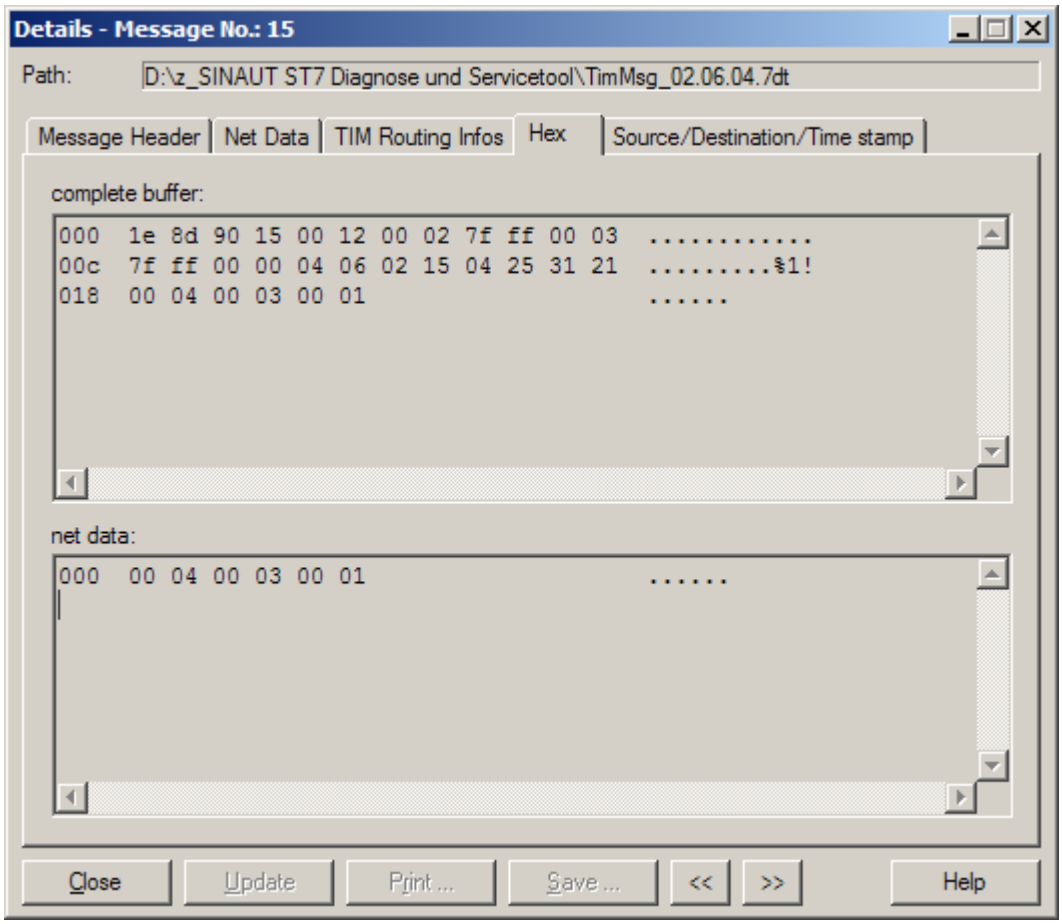


Figure 3-40 *Details* dialog, *Hex* tab

- The *Source/Destination/Time stamp* tab provides information in its 3 in areas on the source, and destination and time stamp of the message. The fields provide the following information:
  - *Source*: Information on the subscriber number, name, type name, station name, object number and index number
  - *Destination*: Information on the subscriber number, name, type name, station name and object number
  - *Time stamp*: Information on the date, time, status, status info.

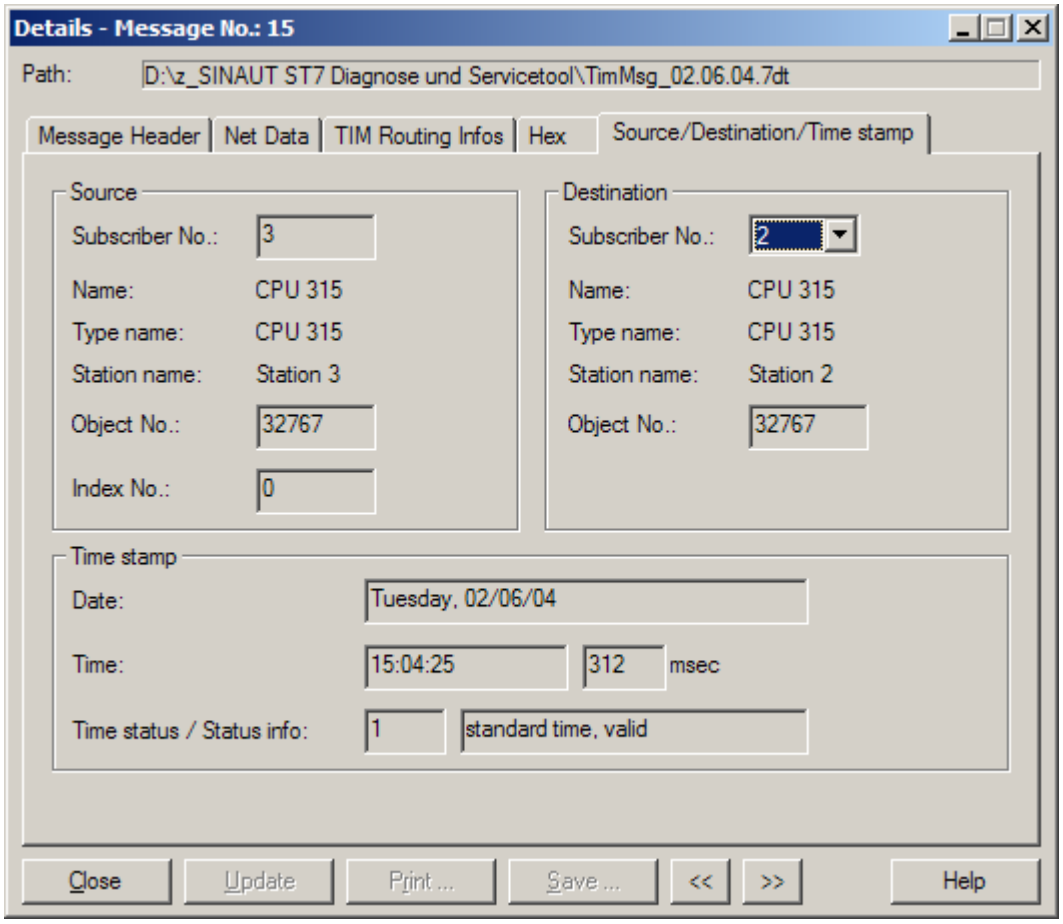


Figure 3-41 *Details* dialog, *Source/Destination/Time stamp* tab

Statistics

The *Statistics* function provides a statistical evaluation of the entire message protocol in terms of numbers, types, and throughput of messages of the subscribers involved assorted according to

- All messages, data messages and organizational messages and according to
- Requested and spontaneous messages.

With the aid of the statistics, you can, for example recognize particular concentrations of certain message types with individual subscribers allowing you to decide whether normal or acceptable message traffic is possible in the particular installation.

The *Statistics* dialog displays the statistical data of the TIM message protocol in three tabs. In each tab, the sampling period of message recalling is displayed at the top. The three tabs of the dialog list the messages as tables sorted according to the following:

- The *Counters* tab provides information on the total number of messages and the number of different sent and received message types.
- The *Message flow* tab provides information on the amount of message traffic per minute. It shows the total number of messages and the number of different sent and received message types per minute.
- The *Subscriber* tab displays a table with the number of different message types per subscriber. With the list box at the top right, you can sort the messages according to subscriber number or message type.

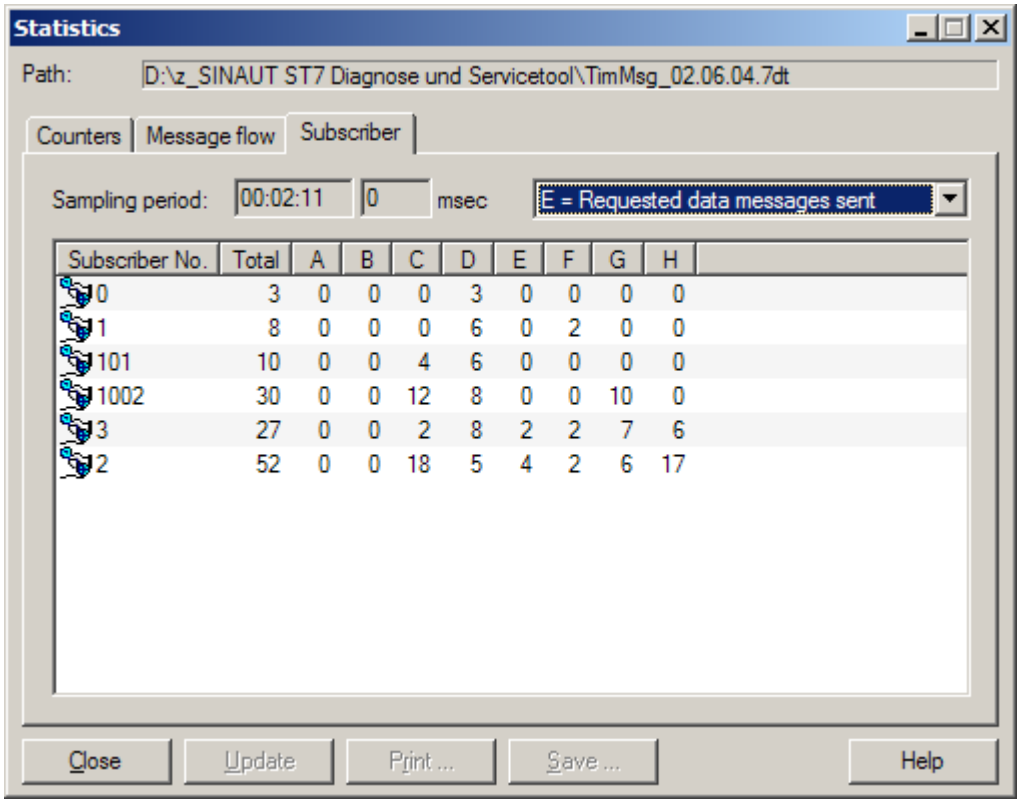

Statistics
Path:
D:\z_SINAUT ST7 Diagnose und Servicetool\TimMsg_02.06.04.7dt
Counters
Message flow
Subscriber
Sampling period:
00:02:11
0
msec
E = Requested data messages sent
Subscriber No. Total A B C D E F G H
0 3 0 0 0 3 0 0 0 0
1 8 0 0 0 6 0 2 0 0
101 10 0 0 4 6 0 0 0 0
1002 30 0 0 12 8 0 0 10 0
3 27 0 0 2 8 2 2 7 6
2 52 0 0 18 5 4 2 6 17
Close
Update
Print ...
Save ...
Help

Figure 3-42 *Statistics* dialog, *Subscriber* tab

The *Statistics* function is only available for individual message protocols (sources) and is not supported, for example, a Testcopy DB or ST7cc protocol was inserted in a TIM message protocol.

Filter functions

Delete list

If you select the *Delete list* function with the mouse, the TIM messages displayed in the list view are *deleted*. This deletion can be reversed. The messages can be inserted in the list view again using the to functions *Selection* and *Show all messages*.

Selection

The *Selection* function is used to select certain message types to be displayed again after they were deleted from the list view. Messages can be selected, for example, according to individual subscribers, direction, message header entries etc. This function allows you to reduce the number of messages to make the list view clearer.

Show all messages

The *Show all messages* function insert or messages of the TIM message protocol in the list view that were previously deleted completely with the *Delete list* function or selectively with the *Selection* function.

Exporting protocol files

Save current list
The *Save current list* function is used to save the currently open list with all data of the current list view in a CSV file. The CSV file can then be read in MS EXCEL.

If you activate the function, an input box opens in which you have the option of entering a comment on the protocol you are saving. This comment is then included in the top rows of the CS the saved file. You then specify the directory and file name of the CSV file.

Export complete list
The *Export complete list* function is used to save the entire country open list with *all* available data in an MS EXCEL-compatible CSV file.

If you activate the *Export complete list* function, an input box opens in which you have the option of entering a comment on the protocol you are saving. This comment is then included in the top rows of the CS the saved file. You then specify the directory and file name of the CSV file.

Note

You can open the CSV file created with the *Save current list* or *Export current list* functions in MS EXCEL by selecting the *File / Open...* menu, so that the data from the individual columns of the list view of the TIM message protocol are shown in separate columns in EXCEL.
If you open the CSV file by double-clicking in the Explorer, the data is shown with separators but nevertheless in one single column.

3.6 Messages in the diagnostic buffer of the TIM

Introduction

In much the same way as on and S7 CPU, a diagnostic buffer is also maintained on the TIM. The TIM stores its specific diagnostic messages in this buffer. The diagnostic messages of the TIM module are read out in the same way as those of a CPU.

Note

If there is no text file with the diagnostic texts of the TIM events on the PG with which the diagnostic buffer is read out, the events are displayed in hexadecimal format.

3.6.1 Diagnostic messages of the TIM

Classification of TIM messages

The TIM uses a reserved area within the event class F, namely Fx60, known as the event ID. All TIM diagnostic messages start with Fx60 in the hexadecimal representation, where x is is the placeholder for an identifier that allows a global classification of the message:

Table 3- 18 Global classification of the TIM diagnostic messages

ID x	Resulting event ID	Classification
2	F260	Event message, exiting state
3	F360	Event message, entering state
4	F460	Event message, internal error, exiting state
5	F560	Event message, internal error, entering state
8	F860	Event message, external error, exiting state
9	F960	Event message, external error, entering state

The event ID Fx60 is followed by the actual message, the detailed event. This occupies the numeric range from 0000h to 0FFFh. Depending on the message, there may be additional information under *Additional info 1/2/3* or *Additional info 4/5*.

The diagnostic messages of the TIM in hexadecimal and plain text format

The following table lists all the TIM diagnostic messages in ascending order of the detailed event in hexadecimal format.

To complete the picture, the corresponding event ID is also listed. The event ID is used only to classify the message and has no relevance for the order.

Table 3- 19 Diagnostic messages of the TIM

Event ID (hex)	Detailed event (hex)	Meaning
General messages		
F560	0001	Entering state: Heap memory overflow.
F460		Exiting state: Heap memory overflow eliminated.
Init task messages		
F360	0060	ST7 installation started.
F560	0061	Error creating the message queue of the INIT task.
F360	0062	Archive was created.
F560	0063	Error installing the DNA interrupt service routine.
F560	0064	Error installing the interrupt service routine for the external interface.
F560	0065	Error installing the interrupt service routine for the internal interface.
F560	0066	Error installing the TIMER interrupt service routine.
F560	0067	Structure of the interface administration could not be entered in the catalog.
F560	0068	Dongle flag not found in catalog.
F560	0069	EXE loader not found in catalog.
F360	006A	Internal / external WAN interface: Driver not released.
F560	006B	Internal / external WAN interface: Error sending the load job for a driver.
F560	006C	Internal / external WAN interface: Error loading the basic task of a driver.
F560	006D	Internal / external WAN interface: Error creating the basic task of a driver.
F560	006E	Internal / external WAN interface: Error receiving the task ID of a driver.
F560	006F	Internal / external WAN interface: Error starting the basic task of a driver.
F560	0070	WAN SDB could not be opened.
F560	0071	WAN SDB could not be found.
F560	0072	Error sending the load job for the clock driver.
F560	0073	Error loading the basic task of the clock driver.
F560	0074	Error creating the basic task of the clock driver.
F560	0075	Error receiving the task ID of the clock driver.
F560	0076	Error starting the basic task of the clock driver.
F560	0077	WAN SDB does not start with the *TIM* parameter block.
F560	0078	Not enough memory available.
F560	0079	Error creating the global message memory.
F560	007A	Error sending the load job for the routing program.
F560	007B	Error loading the basic task of the routing program.
F560	007C	Error creating the basic task of the routing program.
F560	007D	Error receiving the task ID of the routing program.
F560	007E	Error creating the main task of the routing program.
F560	007F	Routing tables were not created within the specified time.
F560	0080	WAN driver was not installed within the specified time.
F560	0081	Unknown parameter block in WAN SDB.
F360	0082	Startup of module completed.
F560	0083	Flag group unknown.

Event ID (hex)	Detailed event (hex)	Meaning
F360	0084	RAM drive was created.
F560	0085	Error creating RAM drive.
F560	0086	Error in memory analysis in HEAP.
F560	0087	Error installing the clock driver. No message received at end of installation.
F360	0088	Message of the message and HEAP memory initialized on the TIM.
F360	0089	Routing SDB could not be found.
F560	008A	Routing SDB could not be opened.
F560	008B	Routing SDB starts with incorrect subnet block ID.
F560	008C	No memory available for routing function.
F560	008D	Incorrect block ID detected in routing SDB.
F560	008E	Error sending a job to the LAN task.
F560	008F	Installation of the routing function was aborted.
F560	0090	ST7 installation started.
F360	0091	Wrong TIM firmware loaded.
F560	0100	Module startup: Installation error
LAN messages		
F560	0101	Installation of the AMPLUS-L emulation aborted.
F560	0103	LAN communication: Error sending a message.
F560	0104	LAN communication: Error receiving a message or ID unknown.
F360	0105	LAN communication: Unknown job.
F360	0106	LAN communication: Connection could not be entered in routing table.
F360	0107	LAN communication: Connection could not be deleted in routing table.
F560	0108	LAN communication: Error in a connection SDB.
F360	0109	LAN communication: No resources available for connection.
F560	010A	LAN communication: PBC connection could not be established. Reference number unknown.
F560	010B	LAN communication: PBC connection could not be established. Reference number unknown.
F360	010C	Entering state: LAN communication: Connection down.
F260		Exiting state: LAN communication: Connection OK.
F560	010D	Entering state: LAN communication: Threat of send queue overflow for a connection.
F460		Exiting state: LAN communication: Threatening send queue overflow for a connection eliminated.
F560	010E	Entering state: LAN communication: Send queue overflow for a connection.
F460		Exiting state: LAN communication: Send queue overflow for a connection eliminated.
F560	010F	Entering state: Disruption of MPI/party line interface (SPC/2) detected.
F460		Exiting state: Disruption of MPI/party line interface (SPC/2) eliminated.
F560	0110	LAN communication: Error reading the LAN SDB.
F560	0111	LAN communication: Error occurred during PBC send.
F560	0112	LAN communication: Error in SDB0 – bad MPI parameter.
F560	0113	LAN communication: A negative acknowledgment was sent.
F560	0114	No Ethernet SDB.
F560	0115	Ethernet SDB could not be opened.

Event ID (hex)	Detailed event (hex)	Meaning
F560	0116	Ethernet SDB with bad block ID received.
F560	0117	Error creating a socket.
F560	0118	Error linking a socket.
F560	0119	Invalid socket.
F560	011A	Error listening on a socket.
F560	011B	Ethernet port: RFC1006 has received a packet whose length exceeds the maximum.
F560	011C	Bad RFC1006 PDU header.
F560	011D	Undefined PDU received.
F560	011E	Bad TCP/IP packet.
F560	011F	Error setting a socket.
F560	0120	Error in ACCEPT socket.
F560	0121	TCP/IP connection termination by partner.
F560	0122	TCP/IP reception error.
F560	0123	TCP/IP send error.
F560	0124	TCP/IP connection number invalid.
F560	0125	Error receiving a CR-PDU.
F560	0126	Illegal access over TCP/IP.
F560	0127	Invalid PDU length.
F560	0128	KEEPALIVE expired.
F560	0129	Entering state: Connection information: Ethernet port problem.
F460		Exiting state: Connection information: Ethernet port ok.
F560	012A	Error in socket CONNECT job.
F560	012B	Error receiving a CC-PDU.
F560	012C	Unknown error code.
F560	012D	Maximum number of S7 connections exceeded.
WAN messages		
F560	0300	Internal / external WAN interface: Installation of the WAN driver was aborted.
F360	0301	Internal / external WAN interface: WAN driver is installed.
F360	0302	Internal / external WAN interface: Connection to a subscriber established (incoming call; subscriber number identified).
F560	0303	Bad organizational message from routing task.
F360	0304	Entering state: Internal / external WAN interface: Send buffer changed over to image.
F260		Exiting state: Internal / external WAN interface: Send buffer changed back from image.
F360	0305	Entering state: Internal / external WAN interface: Send buffer overflow occurred.
F260		Exiting state: Internal / external WAN interface: Send buffer overflow eliminated.
F960	0306	Bad message: max. number of destination subscribers exceeded.
F960	0307	Bad message: Unknown source subscriber number.
F960	0308	Bad message: S7 PDU not with AE ID = 2.
F960	0309	Entering state: Internal / external WAN interface: CTS disturbance occurred on modem.
F860		Exiting state: Internal / external WAN interface: CTS disturbance eliminated on modem.
F560	030A	Entering state: Internal / external WAN interface: USART error occurred.

Event ID (hex)	Detailed event (hex)	Meaning
F460		Exiting state: Internal / external WAN interface: USART error eliminated.
F360	030B	Entering state: Subscriber failed.
F260		Exiting state: Subscriber OK.
F360	030C	Entering state: Internal / external WAN interface: WAN driver disabled.
F260		Exiting state: Internal / external WAN interface: WAN driver enabled.
F360	030D	Entering state: Internal / external WAN interface: Subscriber call disabled.
F260		Exiting state: Internal / external WAN interface: Subscriber call enabled.
F360	030E	Entering state: Internal / external WAN interface: Subscriber call in subcycle.
F260		Exiting state: Internal / external WAN interface: Subscriber call in main cycle.
F360	030F	Entering state: Internal / external WAN interface: Permanent call to a subscriber enabled.
F260		Exiting state: Internal / external WAN interface: Permanent call to a subscriber ended.
F360	0310	Entering state: Internal / external WAN interface: Lack of resources on a subscriber.
F260		Exiting state: Internal / external WAN interface: Lack of resources on a subscriber eliminated.
F360	0311	Entering state: Internal / external WAN interface: Alternative path changeover on.
F260		Exiting state: Internal / external WAN interface: Alternative path changeover off.
F560	0312	Internal / external WAN interface: No message memory available for new image element.
F560	0313	Internal / external WAN interface: Image element too large for image memory.
F560	0314	Internal / external WAN interface: Set number of subscribers exceeded in image.
F560	0316	Internal / external WAN interface: 'List of Active Stations' (LAS) not available.
F560	0317	Internal / external WAN interface: Error receiving a message.
F560	0318	Internal / external WAN interface: Error enabling message memory.
F560	0319	Entering state: Internal / external WAN interface: Communication with AMPLUS-L task disrupted.
F460		Exiting state: Internal / external WAN interface: Communication with AMPLUS-L task OK.
F560	031A	Entering state: Internal / external WAN interface: Communication with clock driver disrupted.
F460		Exiting state: Internal / external WAN interface: Communication with clock driver OK.
F560	031B	Entering state: Internal / external WAN interface: Communication with routing task disrupted.
F460		Exiting state: Internal / external WAN interface: Communication with routing task OK.
F560	031C	Internal / external WAN interface: Modem command invalid.
F560	031D	Internal / external WAN interface: Invalid dialing string or bad call number transferred to modem when calling a subscriber.
F960	031E	Internal / external WAN interface: Incorrect handshake PDU received from a subscriber.
F360	031F	Internal / external WAN interface: Own telephone connection occupied.
F960	0320	Internal / external WAN interface: Modem not replying.
F960	0321	Internal / external WAN interface: Access to called subscriber not permitted.
F960	0322	Internal / external WAN interface: No answer tone received from modem of called subscriber.
F960	0323	Internal / external WAN interface: Called subscriber is not operational.
F960	0324	Internal / external WAN interface: Modem of called subscriber has no power.
F960	0325	Internal / external WAN interface: Telephone line is disrupted.
F360	0326	Internal / external WAN interface: Supervision time exceeded. Repetition starting.

Event ID (hex)	Detailed event (hex)	Meaning
F360	0327	Internal / external WAN interface: All attempts to dial a subscriber were executed. No connection was established.
F360	0328	Entering state: Internal / external WAN interface: Telephone number list with telephone number(s) of a subscriber deactivated.
F260		Exiting state: Internal / external WAN interface: Telephone number list with telephone number(s) of a subscriber activated.
F560	0329	Internal / external WAN interface: Telephone number list with telephone number(s) of a subscriber is invalid or disrupted.
F360	032A	Internal / external WAN interface: Telephone number of a subscriber temporarily disabled.
F360	032B	Internal / external WAN interface: Telephone number of a subscriber was changed.
F560	032C	Internal / external WAN interface: STA number not found in telephone number list.
F360	032D	Entering state: Internal / external WAN interface: Permanent connection established to a subscriber.
F260		Exiting state: Internal / external WAN interface: Permanent connection to a subscriber was terminated.
F360	032E	Entering state: Internal / external WAN interface: Permanent connection to a subscriber was registered.
F260		Exiting state: Internal / external WAN interface: Permanent connection to a subscriber was deregistered.
F560	032F	Internal / external WAN interface: Permanent connection to a subscriber was aborted.
F360	0330	Entering state: Internal / external WAN interface: Incoming call disabled.
F260		Exiting state: Internal / external WAN interface: Incoming call enabled.
F360	0331	Internal / external WAN interface: Establishing connection to a subscriber.
F360	0332	Internal / external WAN interface: Connection to a subscriber established (outgoing call).
F360	0333	Internal / external WAN interface: Connection to a subscriber established (incoming call; subscriber number not yet identified).
F360	0334	Internal / external WAN interface: Connection to a subscriber was terminated.
F560	0335	Internal / external WAN interface: Connection to a subscriber was aborted.
F360	0336	Internal / external WAN interface: Connection to a subscriber is already terminated.
F360	0337	Internal / external WAN interface: Supervision time exceeded. No repetition.
F360	0338	Internal / external WAN interface: Send buffer was deleted.
F360	0339	Internal / external WAN interface: Image memory and send buffer were deleted.
F360	033A	Internal / external WAN interface: No telephone number in modem memory.
F960	033B	Internal / external WAN interface: PDU received with unknown STA number.
F360	033C	Entering state: Internal / external WAN interface: Driver redundancy - memory management switched over.
F260		Exiting state: Internal / external WAN interface: Driver redundancy - memory management switched back.
F560	033D	Internal / external WAN interface: Incorrect service request to pager (SMS).
F360	033E	Internal / external WAN interface: No entries in the send queue.
F560	033F	Internal / external WAN interface: Unknown message type.
F560	0340	Internal / external WAN interface: Communication with WAN driver disrupted.
F360	0341	Internal / external WAN interface: Incoming call (RING).

Event ID (hex)	Detailed event (hex)	Meaning
F960	0342	Internal / external WAN interface: No connection with incoming call.
F960	0343	Internal / external WAN interface: Call or connection abort.
F960	0344	Internal / external WAN interface: No carrier frequency detected on partner.
F360	0345	Entering state: Internal / external WAN interface: Further dialing attempts were made in the background to a disturbed subscriber.
F260		Exiting state: Internal / external WAN interface: Subscriber is available again. Dialing attempts in the background will be stopped.
F960	0346	Internal / external WAN interface: Incorrect PIN number transferred to GSM module.
F960	0348	Internal / external WAN interface: Error occurred in GSM module.
F960	0349	Internal / external WAN interface: GSM module not responding or not available.
F960	034A	Internal / external WAN interface: SMS server of the TIM has received an unknown message.
F960	034C	Internal / external WAN interface: Short message (SMS) acknowledgment received from an unknown mobile subscriber.
F360	034E	Internal / external WAN interface: Incoming call detected. Incoming calls are disabled.
F360	034F	Internal / external WAN interface: Incoming call detected. DTR signal was activated.
F360	0350	Internal / external WAN interface: SMS server of the TIM was installed and started.
F360	0351	Internal / external WAN interface: SMS status, global status request/deletion.
F360	0352	Internal / external WAN interface: SMS status, single status request/deletion.
F360	0353	Internal / external WAN interface: Spontaneous SMS status message.
F960	0354	Internal / external WAN interface: Short message (SMS) acknowledgment incorrect. Format or ID no. unknown.
F560	0355	Internal / external WAN interface: No send buffer could be made available for sending a short message (SMS).
F960	0356	Internal / external WAN interface: The GSM module expects the PUC number.
F960	0357	Internal / external WAN interface: Telephone number of the SMS recipient could not be found.
F360	0358	Internal / external WAN interface: The GSM signal strength is xx dBm.
F360	0359	Subscribers cannot be blocked.
F360	035A	Internal / external WAN interface: GSM module detects wrong service ID.
F360	035B	Internal / external WAN interface: GSM module ready to receive.
F360	035C	Internal / external WAN interface: The switchover to the image method for blocked messages was forced.
F360	035D	Internal / external WAN interface: Threat of forced switchover to image method.
F360	035E	Internal / external WAN interface: Threat of forced switchover to image method.
F360	035F	Internal / external WAN interface: Permanent connection already active.
F360	0360	Internal / external WAN interface: Maximum number of messages exceeded.
F360	0361	Entering state: Internal / external WAN interface: The data brake for the connection to a subscriber was enabled.
F260		Exiting state: Internal / external WAN interface: The data brake for the connection to a subscriber was disabled.
F960	0362	Internal / external WAN interface: After transferring the PIN to the GSM module, no network contact could be established.
Messages from the routine task		
F560	0500	Installation of the routing program aborted.

Event ID (hex)	Detailed event (hex)	Meaning
F560	0501	Internal / external WAN interface: Receive task of the WAN driver unknown.
F560	0502	Receive task of the clock driver unknown.
F560	0503	Read time function unknown.
F560	0504	Receive task of the LAN task unknown.
F560	0505	Partner table unknown.
F560	0506	Error occurred receiving a message.
F560	0507	Unknown PDU received.
F560	0508	PDU with bad address received. Destination subscriber number not found.
F560	0509	No WAN driver available.
F560	050B	Error enabling heap memory.
F560	050C	Installation of the routing program ended. All routing tables available.
F560	050D	Error occurred sending a message.
F560	050E	No resources for creating the destination address table.
F560	050F	Destination address table not created.
F560	0510	No resources for PDU copy.
F560	0511	No resources for copy of partner table.
F360	0512	Entering state: Internal / external WAN interface: Redundancy function activated.
F260		Exiting state: Internal / external WAN interface: Redundancy function deactivated.
F960	0513	Message with incorrect block length received or block length is zero.
F560	0514	Max. number of messages exceeded.
F560	0515	Error in time-of-day synchronization over LAN.
F560	0516	MesA - Error sending a message.
F560	0517	The partner table/substitute table of a subscriber does not exist.
F560	0518	Error releasing memory.
F560	0519	Error in the MesA memory management
F560	051A	Entering state: MesA - Start of indication of a message memory overflow.
F460		Exiting state: MesA - End of indication of a message memory overflow.
F560	051B	Entering state: MesA - Start of indication of a message memory overflow.
F460		Exiting state: MesA - End of indication of a message memory overflow.
F560	051C	Message memory overflow occurred.
F560	051D	Entering state: Start of indication of a message buffer overflow.
F460		Exiting state: End of indication of a message buffer overflow.
F560	051E	Entering state: Overflow of message buffer active.
F460		Exiting state: Overflow of message buffer deactivated.
F560	051F	Buffer with messages was deleted.
F560	0520	MesA - Error in the dynamic assignment table.
F560	0521	MesA - WAN/LAN router: Error sending messages.
F560	0522	MesA could not send the acknowledgment for an org. 262 PDU to SubA.
F560	0523	MesA - WAN/LAN router; error sending over WAN/LAN driver.
F560	0524	Installation error occurred in the MesA.
F560	0530	MesA - System error occurred.

Event ID (hex)	Detailed event (hex)	Meaning
F560	0531	MesA - Error in the request for an Org4/14 message.
F560	0532	MesA - Unknown control command received.
F560	0533	MesA - Error in MesA system status list query.
F560	0534	MesA: Bad PDU detected.
Messages from clock driver		
F560	0620	Installation of clock driver aborted.
F360	0621	for error code = 0: Installation of the clock driver completed for error code = 1: Installation of clock driver aborted.
F560	0622	Entering state: Time synchronization (master) disturbed.
F460		Exiting state: Time synchronization (master) OK.
F960	0623	Entering state: Time synchronization (slave) disturbed.
F860		Exiting state: Time synchronization (slave) OK.
F560	0624	Error occurred setting the RMOS clock.
F560	0625	Error occurred reading the RMOS clock.
F360	0626	Illegal setting of the RMOS clock by *Set clock* PG service. TIM has onboard DCF77 clock.
F360	0627	Unknown message received.
F360	0628	Time synchronization PDU with incorrect ORG number received.
F360	0629	Bad synchronization PDU received from LAN.
F560	062A	Entering state: Time synchronization by DCF77 clock disturbed.
F460		Exiting state: Time synchronization by DCF77 clock OK.
F960	062B	Entering state: DCF77 clock radio signal disturbed.
F860		Exiting state: DCF77 clock radio signal OK.
F360	062C	TIM time OK. RMOS clock set for the first time.
F360	062D	Entering state: Changeover to daylight saving time.
F260		Exiting state: Changeover to standard time.
F360	062E	Entering state: Notification hour for daylight saving/standard time changeover active.
F260		Exiting state: Notification hour for daylight saving/standard time changeover completed.
F360	0630	Change in synchronization mode.
F360	0631	Daylight saving/standard time changeover performed manually.
F360	0632	Error occurred in manual daylight saving/standard time changeover.
F560	0633	Error occurred starting the synchronization task.
F560	0634	Error occurred starting the control task for synchronization.
F960	0635	Two time masters detected in one network.
F960	0636	Incorrect synchronization mode on MPI. Master mode expected.
F960	0637	No further module exists on MPI bus/party line.
F560	0638	External / internal WAN interface: A subscriber could not be synchronized following restart.
F360	0639	Error detected in time-of-day synchronization.
F360	063A	Bad time-of-day message received from DCF77 module.
F360	063B	Bad time-of-day message received from DCF77 module.
F360	063C	Time jump occurred.
F360	063D	Bad time-of-day message received from DCF77 module.

Event ID (hex)	Detailed event (hex)	Meaning
F560	063E	Error in synchronization request.
Messages from diagnostic server		
F560	0660	Error occurred installing the diagnostic server.
F360	0661	Installation of diagnostic server completed.
F560	0662	Error occurred sending a message.
F560	0663	Entering state: Error occurred receiving a message.
F460		Exiting state: Error occurred receiving a message.
F560	0664	Entering state: Heap memory overflow.
F460		Exiting state: Heap memory overflow eliminated.
F960	0665	Entering state: Unknown PDU received.
F860		Exiting state: Unknown PDU received.
F560	0666	Incorrect firmware version installed on TIM.
F360	0667	Entering state: All [n] LAN connections are disrupted.
F260		Exiting state: [x] of [n] LAN connections are OK.
F360	0668	Message buffer of TIM records was deleted.
F360	0669	Entering state: ST7 Message Monitor on
F260		Exiting state: ST7 Message Monitor off
F360	066A	Entering state: Extended diagnostics on
F260		Exiting state: Extended diagnostics off
F360	066B	Extended diagnostics - modification
F360	066C	Error occurred in system status list query.
F560	066D	Wrong firmware.
Messages from P bus server		
F560	06B0	Installation of P bus server aborted.
F360	06B1	Unknown message received from task.
F560	06B2	Error occurred receiving a message.
F360	06B3	Entering state: Power outage on P bus.
F260		Exiting state: Power supply on P bus OK.
F360	06B4	Entering state: I/O disabled by CPU.
F260		Exiting state: I/O enabled by CPU.
F560	06B5	Module on P bus not capable of communication.
F960	06B6	Parity error in P bus communication.
F960	06B7	Bit shift error in P bus communication.
F360	06B8	Diagnostic interrupt cannot be sent. Module is not enabled on P bus.
F360	06B9	Diagnostic interrupt cannot be sent. Diagnostic interrupt is not enabled on P bus.
F360	06BA	SDB0.SDB cannot be opened. File overwritten without comparison.
F360	06BB	SDB0.SDB cannot be created.
F360	06BC	SDB0.SDB cannot be written.
F560	06BD	BUS3-ASIC could not be initialized.
F360	06BE	SDB5.SDB cannot be opened. File overwritten without comparison.
F360	06BF	SDB5.SDB cannot be created.

Event ID (hex)	Detailed event (hex)	Meaning
F360	06C0	SDB5.SDB cannot be written.
Messages from the SDB handler		
F560	06E0	Installation of the SDB handler aborted.
F560	06E1	SDB could not be copied.
F560	06E2	SDB could not be deleted.
F560	06E3	SDB could not be loaded.
F560	06E4	SDB information of hierarchy 1 not available.
F560	06E5	SDB information of hierarchy 2 not available.
F560	06E6	SDB information of hierarchy 3 not available.
F560	06E7	Error chaining SDBs.
F360	06E8	SDB handler: Unknown job.
F360	06E9	Control instruction unknown.
F560	06EB	Error occurred during firmware update.
TD7onTIM messages		
F560	0700	Entering state: TD7 installation started.
F460		Exiting state: TD7 installation ready.
F560	0701	Semaphores not created.
F560	0702	Semaphores not created.
F560	0703	SecIntervall task was not started.
F560	0704	TD7_ObjectAdmin task was not started.
F560	0705	The path for SDB files could not be opened.
F560	0706	The TD7 SDB could not be found.
F560	0707	The TD7 SDB could not be opened.
F560	0708	Header with incorrect length of block ID.
F560	0709	TD7-SDB: No TD7 parameters found.
F560	070A	TD7-SDB: Block not found.
F560	070B	TD7-SDB: T4T_MAINHDR has incorrect length.
F560	070C	TD7-SDB: T4T_SUBDATA has incorrect length.
F560	070E	TD7-SDB: Unknown format in a destination subscriber block.
F560	0711	TD7-SDB: Number of partner blocks incorrect.
F560	0712	TD7-SDB: Unknown format in a partner block.
F560	0713	TD7-SDB: Unknown channel type.
F560	0715	TD7-Run: Not enough memory for the channel list.
F560	0716	TD7-SDB: Channel block not found.
F560	0717	TD7-Run: Not enough memory for a channel object.
F560	0718	TD7-SDB: The number of data entries is incorrect.
F560	071B	TD7-SDB: The number of object entries is incorrect.
F560	071C	TD7-Par: Subscriber object for a partner not found.
F560	071D	TD7-Par: Unknown partner.
F560	0721	TD7-Par: Invalid scan cycle ID.
F560	0722	TD7-Run: Object not in fast cycle.

Event ID (hex)	Detailed event (hex)	Meaning
F560	0725	TD7-SDB: No objects found in header ID.
F560	0726	TD7-SDB: Unknown format in an object.
F560	0729	TD7-Par: Not enough memory for scan cycle job list.
F560	072A	Basic channel memory assignment error.
F560	072D	TD7-Par: Wrong channel data type.
F560	072E	Unknown channel type.
F560	0730	TD7-Par: Invalid scan cycle ID.
F560	0731	Memory assignment error creating the object list for the current subscriber.
F560	0732	TD7-SDB: Number of objects does not match the number of objects in the header.
F560	0733	A read job to the CPU was not responded to after 1 ms.
F560	0736	TD7-Com: CPU communication error in object X, channel Y.
F560	0737	TD7-Com: CPU access error for object X, channel Y.
F560	073B	Memory assignment error creating the message buffer for a scan cycle.
F560	073D	A write job to the CPU was not responded to after 1 ms.
F560	073E	TD7-Com: Negative acknowledgment from LAN communication for job from scan cycle.
F560	0742	TD7-Run: Error reading an input trigger.
F560	0744	TD7-Par: Object without channels.
F560	074A	TD7-Par: Invalid address with trigger signal.
F560	074B	TD7-Par: Invalid address for net data in object X, channel Y.
F560	074C	An invalid address was reported for object X in channel Y.
F560	0752	TD7-SDB: Number of subscriber blocks does not match main header entry.
F560	0755	Unknown channel type.
F560	0759	Not enough memory to create the TD7onTIM send job list.
F560	0760	Initialization of TD7onTIM for source subscriber complete.
F560	0761	The general request of a subscriber is incomplete.
F560	0762	Timeout in the general request to a subscriber.
F560	0763	The general request of object X of a destination subscriber is incomplete.
F560	0766	Message with unknown source subscriber.
F560	0767	Unknown start index in received organizational message.
F560	0768	TD7-Run: Received organizational message not accepted due to invalid length.
F560	076C	Invalid start index in received data message.
F560	076D	TD7-Run: Received data too large for destination object.
F560	076E	TD7-Run: Start index of received data message does not match the receive channel of the destination object.
F560	0773	TD7-Run: Object without partner or channels.
F560	0774	TD7-Run: No destination object found.

Diagnostic messages and activation of the group error LED

If an error occurs during startup, the read group LED (SF) of the TIM lights up and a message to this effect is entered in the diagnostic buffer of the TIM.

If the TIM is installed as a CP in an S7-300 rack, a diagnostic interrupt is sent to the CPU.

The following table contains a summary of all error messages that caused the group error LED (SF) to light up.

Table 3- 20 Classification of the group error messages of the TIM

Error class	Detailed event in the diagnostic message
Internal error	0061h 0063h - 0067h 0069h 0078h - 0080h 0083h 06B0h
External error	0320h
No parameter assignment:	0070h 0071h 0620h
Bad parameter assignment	0077h 0081h 0110h 0112h 0300h 031C 0500h 0620h
RAM error	0085h 0086h

3.7 Messages in the diagnostic buffer of the CPU

Introduction

The SINAUT TD7onCPU software package generates a series of operating and error messages that are entered in the diagnostic buffer of the CPU. These can be displayed in *SINAUT / TD7 CPU Diagnostics*. If a PG is connected and if this is registered for CP messages, these diagnostic messages are displayed on the PG immediately when they arise.

The messages generated by TD7 are, however, only displayed in plain text when they are called up in *SINAUT / TD7 CPU Diagnostics*. If you call up the messages generated by TD7 in *STEP 7 Diagnostics / Module Information, Diagnostic Buffer tab*, they are displayed in hexadecimal format.

Note

If there is no text file with the diagnostic texts of the TD7 events on the PG with which the diagnostic buffer is read out, the events are displayed in hexadecimal format.
Based on the following list, you can identify the significance.

3.7.1 SINAUT diagnostic messages of TD7onCPU

Classification of TD7 messages

The TD7onCPU software package uses the numeric range of event class B for its diagnostic messages. This event class is reserved for user-defined events in STEP 7. It is, however, used in SINAUT ST7 for SINAUT diagnostic messages.

Note

In SINAUT ST7, the user can only use event class A for user-defined diagnostic messages. Just as in event class B, this is also reserved for user-defined messages in STEP 7.

The SINAUT diagnostic messages of TD7onCPU use the range from Bx00h to BxFFh. To distinguish the individual messages, only in the last two numbers are significant, in other words the range from 00h to FFh.

x is a placeholder for an ID that allows the global classification of messages. For the TD7 diagnostic messages, the IDs 0 and 1 are used for events exiting state (0) and entering state (1).

Table 3- 21 Global classification of the SINAUT diagnostic messages of TD7onCPU

ID x	Resulting event IDs	Classification
0	B000h ... B0FFh	Exiting state messages
1	B100h ... B1FFh	Entering state messages

Most messages are entering state, only a few are existing state.

The TD7 diagnostic messages in hexadecimal and plain text format

The following table shows or SINAUT TD7 diagnostic messages according to their event IDs numbers in ascending order in hexadecimal format.

Most diagnostic messages have additional information added to them. If the diagnostic messages are displayed as text, this additional information is included in the texts suitably formatted.

In the explanations below, this additional information is specified within the texts as [Info1] or [Info2/3]. If the texts are not shown, the additional information is shown as 'Additional info 1 / 2 / 3' in the hexadecimal representation on the PG.

Table 3- 22 SINAUT diagnostic messages of TD7onCPU

No.	Event ID (hex)	Meaning
Parameter assignment errors, configuration errors		
0	B100	Parameter assignment error for object no. [Info1]: PartnerNo [Info2/3] not permitted.
1	B101	Parameter assignment error for object no. [Info1]: PartnerNo [Info2/3] unknown.
2	B102	Parameter assignment error for object no. [Info1]: PartnerObjectNo [Info2/3] not permitted.
3	B103	Parameter assignment error for object no. [Info1]: ST1 message no. [Info2/3] not permitted.
4	B104	Parameter assignment error for object no. [Info1]: ST1 object no. [Info2/3] not permitted.
5	B105	Parameter assignment error for object no. [Info1]: ST1 index no. [Info2/3] not permitted.
6	B106	Parameter assignment error for object no. [Info1]: ST1 PACK value [Info2/3] not permitted.
7	B107	Errors generating the object reference list: DB[Info1] cannot be created. Cause: [Info2/3].
8	B108	Parameter assignment errors in the object reference list: Reference to subscriber no. [Info1] and object no. [Info2/3] exists more than once.
9	B109	Object reference list 1 missing. Searching for reference to subscriber no. [Info1] and object no. [Info2/3].
10	B10A	Object reference list 2 missing. Searching for reference to subscriber no. [Info1] and object no. [Info2/3].
11	B10B	Parameter assignment errors in object reference list 1: Reference to subscriber no. [Info1] and object no. [Info2/3] missing.
12	B10C	Parameter assignment errors in object reference list 2: Reference to subscriber no. [Info1] and object no. [Info2/3] missing.
13	B10D	Error in object DB no. [Info1]: SINAUT ST7 ID A5h missing.
14	B10E	Configuration error: No subscriber contained in DB BasicData (DB[Info1]) .
15	B10F	Unknown connection type for connection no. [Info1] to local subscriber no. [Info2/3].
16	B110	Unknown subscriber type (= [Info1]) for subscriber no. [Info2/3].
17	B111	Data received from an unknown local communication partner (MPI no. = [Info2/3]).
18	B112	In the send buffer of communication DB no. [Info1], an org. message with an unknown destination subscriber (= [Info2/3]) was deleted.
19	B113	Parameter assignment error for FC [Info1] in parameter [Info2/3].
20	B114	Parameter assignment error object no. [Info1] in parameter [Info2/3].

No.	Event ID (hex)	Meaning
Reception errors		
32	B120	Content of receive buffer in communication DB no. [Info1] not plausible: Length of a message greater than the maximum receive length of [Info2/3] bytes.
33	B121	Content of receive buffer in communication DB no. [Info1] not plausible: Length of a message = 0.
34	B122	Content of receive buffer in communication DB no. [Info1] not plausible: Length of all messages greater than length of the receive data block.
35	B123	Received message in communication DB no. [Info1] incorrect: too many destination subscribers (= [Info2/3]).
36	B124	Received message in communication DB no. [Info1] incorrect: Subscriber no. too long (L = [Info2/3]).
37	B125	Org. Received message in communication DB no. [Info1] incorrect: Number of destination subscribers (= [Info2/3]) must be 1.
38	B126	Org. Received message in communication DB no. [Info1] incorrect: Address extension not permitted (AE = [Info2/3]).
39	B127	Received message in communication DB no. [Info1] incorrect: Address extension not permitted (AE = [Info2/3]).
40	B128	Org. Received message in communication DB no. [Info1] incorrect: Destination object no. (= [Info2/3]) not permitted (only 1 ... 32000 or 32767).
41	B129	Org. Received message (ST1) in communication DB no. [Info1] incorrect: Source object no. (= [Info2/3]) not permitted (only 0 or 32767).
42	B12A	Org. Received message for object no. [Info2/3] incorrect: Start index is >< 0. Current communication DB = DB[Info1].
43	B12B	Org. Received message for object no. [Info2/3] incorrect: Number of net received bytes is > 2. Current communication DB = DB[Info1].
44	B12C	Received message for object no. [Info2/3] incorrect: Too many net received bytes. Current communication DB = DB[Info1].
45	B12D	Org. Received message for subscriber object no. [Info2/3] incorrect: Subscriber unknown. Current communication DB = DB[Info1].
46	B12E	Org. received message for subscriber object no. [Info2/3] incorrect: Too many net received bytes. Current communication DB = DB[Info1].
47	B12F	Received message for object no. [Info1] incorrect: Source subscriber no. [Info2/3] unknown.
48	B130	Received message for object no. [Info1] rejected: Source subscriber no. [Info2/3] incorrect.
49	B131	Received message for object no. [Info1] rejected: Source object no. [Info2/3] incorrect.
50	B132	Short Message Service (SMS): Status return message for object no. [Info1] from SMS center subscriber no. [Info2/3] cannot be interpreted.
51	B133	Short Message Service (SMS): Status return message for object no. [Info1] contains unknown status no. [Info2/3]
52	B134	Received message in communication DB no. [Info1] incorrect: Index no. is negative
53	B135	Org. Received message in communication DB no. [Info1] incorrect: The block length [Info2] of the org. section is not plausible.
54	B136	The receive error was detected by the TIM with subsbcriber number [Info1] when reading the data record. TIM error code = [Info2].
55	B137	A receive error occurred when the TIM with subsbcriber number [Info1] was reading the data record. An unexpected status = [Info2] was read in from the TIM I/O.
56	B138	A receive error occurred when the TIM with the subsbcriber number [Info1] was reading the data record. The length [Info2] of the indicated receive block is not permitted.

No.	Event ID (hex)	Meaning
57	B139	A receive error occurred when the TIM with the subsbcriber number [Info1] was reading the data record. The actual data record length differs by [Info2] bytes from the indicated data record length.
58	B13A	A receive error occurred when the TIM with the subsbcriber number [Info1] was reading the data record. Error code (SFC RD_REC) = [Info2].
Send errors		
64	B140	Content of send buffer in communication DB no. [Info1] not plausible. Length of a send message too short: [Info2/3] bytes.
65	B141	Content of send buffer in communication DB no. [Info1] not plausible. Length of a send message too long: [Info2/3] bytes.
66	B142	Content of send buffer in communication DB no. [Info1] not plausible. A sender message longer than the send buffer length: [Info2/3] bytes.
67	B143	Content of send buffer in communication DB no. [Info1] not plausible. Length of a send message odd: [Info2/3] bytes.
68	B144	Content of send buffer in communication DB no. [Info1] not plausible. Sum of all send message lengths incorrect: [Info2/3] bytes.
69	B145	Data message from object no. [Info1] to destination subscriber no. [Info2/3] could not be entered in the communication DB.
70	B146	Organisational message from object no. [Info1] to destination subscriber no. [Info2/3] could not be entered in the communication DB.
71	B147	Short Message Service (SMS): Monitoring time elapsed for SMS message no. [Info1] sent over SMS center subscriber no. [Info2/3]. No repetition.
72	B148	Entering state: Short Message Service (SMS): Monitoring time elapsed for SMS message no. [Info1] sent over SMS center subscriber no. [Info2/3]. Being repeated with backup mobile phone no.
	B048	Exiting state: Short Message Service (SMS): SMS message no. [Info1] sent over SMS center subscriber no. [Info2/3] could be delivered after repetition.
73	B149	Short Message Service (SMS): SMS message no. [Info1] was deleted because object no. [Info2/3] is disabled.
74	B14A	Short Message Service (SMS): Monitoring time elapsed for send job from object no. [Info1] sent to SMS center subscriber no. [Info2/3].
75	B14B	The send error was detected when writing the data record to the TIM with subsbcriber number [Info1]. Error code (SFC WR_REC) = [Info2].
76	B14C	The send error was detected when writing the data record by the TIM with subsbcriber number [Info1]. TIM error code = [Info2].
Connection displays		
80	B150	Entering state: Connection to local subscriber no. [Info1] disrupted.
	B050	Exiting state: Connection to local subscriber no. [Info1] OK.
81	B151	Entering state: Subscriber no. [Info1] disrupted.
	B051	Exiting state: Subscriber no. [Info1] OK.
82	B152	Monitoring time for send job on local subscriber no. [Info1] elapsed. [Info2] = 0: The job could not be completed in time. [Info2] = 1: The job could not be started in time. [Info2] > 1: No. of the communication DB.
83	B153	Send job on local subscriber no. [Info1] failed. The job will be repeated. Communication DB no. = [Info2].
84	B154	Monitoring time for receive job from local subscriber no. [Info1] elapsed. [Info2] = 0: The job could not be completed in time. [Info2] = 1: The job could not be started in time.

No.	Event ID (hex)	Meaning
85	B155	Receive job of local subscriber no. [Info1] failed. The job will be repeated. Communication DB no. = [Info2].
Request displays		
96	B160	General request to subscriber no. [Info1] incomplete.
97	B161	Single request to subscriber no. [Info1] incomplete.
98	B162	Error in general request to object no. [Info1]. Source subscriber no. = [Info2/3].
99	B163	Error in single request to object no. [Info1]. Source subscriber no. = [Info2/3].
Command/setpoint displays		
112	B170	Command input error for object no. [Info1]: Simultaneous input over hardware and software input.
113	B171	Command input error for object no. [Info1]: 1-out-of-8 error.
114	B172	Command/setpoint input error for object no. [Info1]: 1-out-of-n error. Additional info: [Info2/3] (0 = Comm. or Setp.; 1 = Comm.; 2 = Setp.)
115	B173	Command output error for object no. [Info1]: Command and control byte not identical.
116	B174	Command output error for object no. [Info1]: 1-out-of-8 error.
117	B175	Command output error for object no. [Info1]: 1-out-of-n error.
Time displays		
128	B180	Daylight saving/standard time changeover at change of day not permitted.
129	B181	Date/time error when setting the CPU clock. Incorrect date: [Info1].
130	B182	Date/time error when setting the CPU clock. Incorrect time: [Info2/3].
131	B183	Time synchronization disrupted on CPU.
Processing errors		
144	B190	Data loss in object no. [Info1]. Last received data was not processed by typical.
145	B191	Received data loss in subscriber object no. [Info1]. Last received data was not processed by BasicTask.
Operating displays		
160	B1A0	Initialization of subscriber no. [Info1] completed.

SINAUT PG Routing

4

4.1 What is PG Routing?

4.1.1 Introduction

In the SIMATIC world, the term routing is defined as follows:

Routing is finding the path for information beyond network boundaries

In the SIMATIC world at the present time, it is not possible to configure a connection over which data can be transmitted between the two endpoints involved if this connection goes beyond the boundaries of a network.

What is possible, however, is PG routing. Using the PG routing, it is possible to access a programmable module or a module with diagnostic capability beyond network boundaries from a programming device (PG) or computer (PC).

SIMATIC PG routing and SINAUT PG routing

PG routing allows any type of diagnostics with diagnostics-compliant modules. Test, commissioning, and service functions can be executed, such as opening blocks online, monitoring, editing and overwriting or changing the operating mode mode of modules.

SIMATIC PG routing
SIMATIC PG routing is possible only over network types such as MPI, PROFIBUS, and Ethernet. SIMATIC PG routing was released with STEP 7 V5.0.

SINAUT PG routing
Expanding SIMATIC PG routing, SINAUT PG routing also works over SINAUT networks; in other words, in WANs.

In terms of functionality, SIMATIC PG routing and SINAUT PG routing are largely identical with the only difference being that SINAUT PG routing also functions beyond the boundaries of SINAUT networks. This gives the user a convenient option of remote programming and remote diagnostics over the company telecontrol network.

Note

In the remainder of this chapter, the terms *PG routing* and *remote* are used with the following meaning:

- PG routing
  PG routing is PG routing over network types such as MPI, PROFIBUS, and Ethernet as well as SINAUT networks.
- Remote
  A remote CPU or remote TIM is a module that can be accessed from a PG over SINAUT telecontrol networks.

4.1.2 Examples of configuration for PG routing

The following figures illustrate the basic principle of PG routing based on configurations.

Basic configuration of PG routing

In the basic configuration, a PG in the master station is configured and connected to the MPI bus. PG routing extends from the master TIM over a SINAUT network (WAN) to a station TIM over which the CPU connected to it can be reached.

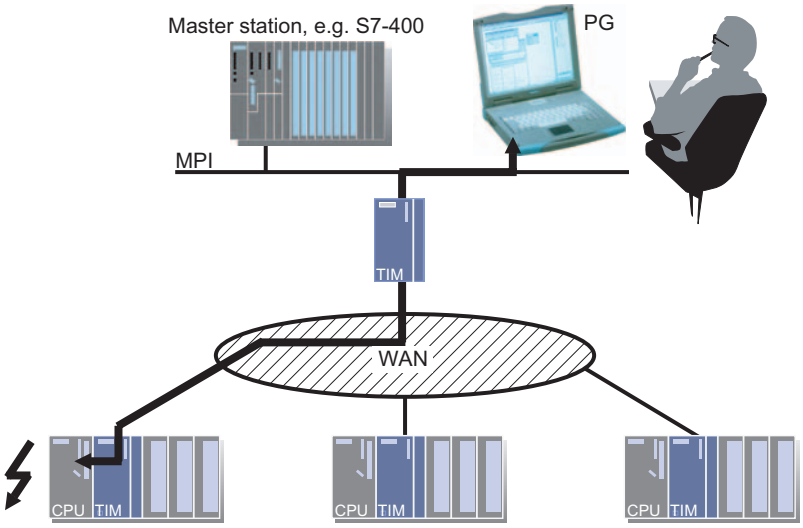


Figure 4-1 Basic configuration of PG routing

PG routing from a SINAUT ST7cc control center with SIMATIC STEP 7

In this case, PG routing does not extend from a PG in the master station but from a central SINAUT ST7cc control system connected to the MPI bus of the master TIM. The routing path shown here is basically the same as described in the basic configuration with a centrally configured PG.

A SINAUT system configuration equipped with SINAUT ST7cc as the central control system already has the full range of functions of PG routing if SIMATIC STEP 7 is also installed alongside the SINAUT ST7cc software. PG routing is then activated in any case. This means that no further preparations are necessary for PG routing to be able to use the parameter assignment, diagnostics, and service functions for remote subscribers over SINAUT networks.

For this configuration, PG routing extends from the PC with the STEP 7 software over the master TIM and a SINAUT network to a SINAUT station where both the station TIM and the CPU can be accessed.

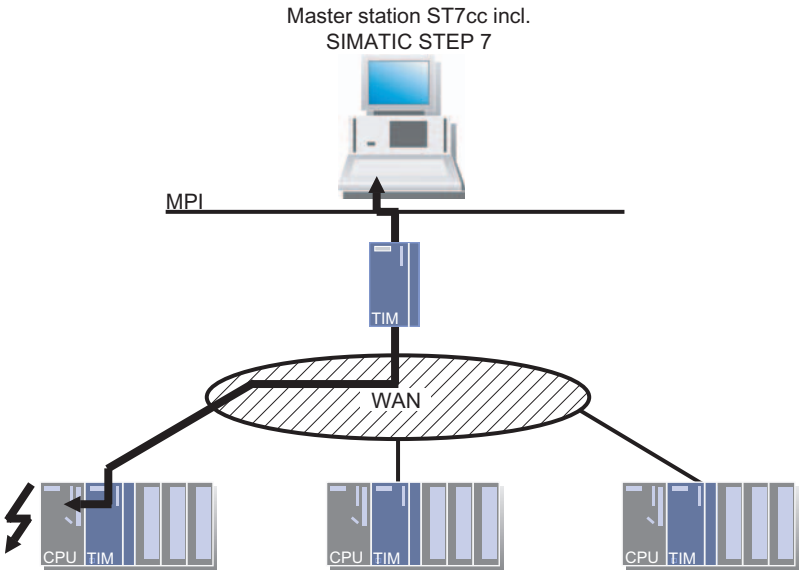


Figure 4-2 PG routing from a SINAUT ST7cc control center with SIMATIC STEP 7

Indirect PG routing over remote PC / laptop with remote access

The configuration below is initially the same as with the central SINAUT ST7cc control system. PG routing, in this case, is used with a remote PC connected over the telephone/ISDN network. With the aid of remote access software, a connection is established to the PC in the master station so that this can be controlled by the remote PC / laptop over the telephone/ISDN connection. From there, PG routing extends over the master TIM and a SINAUT network to a SINAUT station where both the station TIM and the CPU can be accessed.

Various products, such as Symantec pcAnywhere are available on the market and can be used as remote access software.

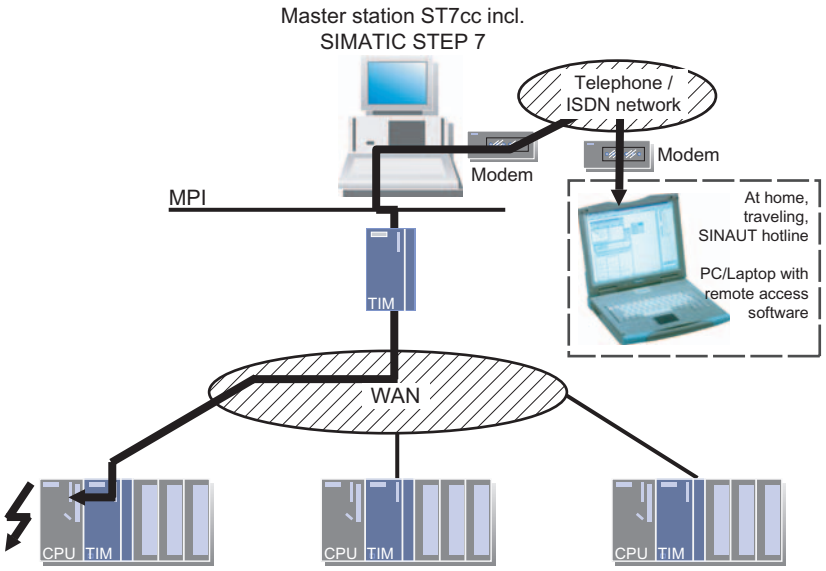


Figure 4-3 Indirect PG routing over a remote PC / laptop with remote access

Indirect PG routing over remote PG/PC with SIMATIC STEP 7

The basic configuration with a configured PG in the master station can be expanded by a teleservice adapter. The PG is configured and connected to the MPI bus in the master station. The teleservice adapter allows the PG to be located remotely and connected over a telephone/ISDN network. As a result, the MPI bus in the master station is extended over a modem connection.

In this case, PG routing extends from the PG with the SIMATIC STEP 7 software package over the modem connection, the master TIM and a SINAUT network to a SINAUT station where both the station TIM and the CPU can be accessed.

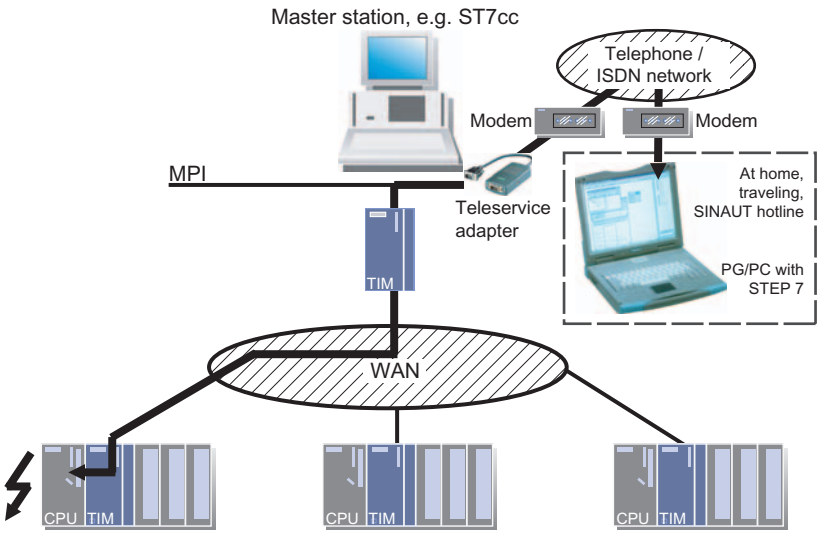


Figure 4-4 Indirect PG routing over a remote PG/PC with SIMATIC STEP 7

4.1.3 Range of functions of PG routing

Range of functions of PG routing

SINAUT ST7 allows configuration of simple and complex networks. For PG routing, this means that the access to remote TIM and CPU modules always follows the hierarchical arrangement of a communication network "from top to bottom"; in other words from a master station to the node station or station.

On the communication link, the following three levels are therefore significant:

Table 4- 1 Communication path of PG routing

Starting point	Intermediate point	Target
PG/PC on the MPI bus of the master TIM	TIM modules in a node station (possibly cascaded)	Station TIM or station CPU

The following table shows the starting point and the communication paths leading to the target with which PG routing functions.

Table 4- 2 Overview of routing-compliant communication paths

Starting point	over	Intermediate point	over	Target
Master station	-	-	D	Station
Master station	-	-	DT	Station
Master station	-	-	DMT	Station
Master station	-	-	DN	Station
Master station	-	-	SPN	Station
Master station	-	-	D	Node station
Master station	-	-	DT	Node station
Master station	-	-	DMT	Node station
Master station	-	-	DN	Node station
Master station	-	-	SPN	Node station
Node station	-	-	D	Station
Node station	-	-	DT	Station
Node station	-	-	DMT	Station
Node station	-	-	DN	Station
Node station	-	-	SPN	Station
Master station	D	Node station	D	Station
Master station	D	Node station	DN	Station
Master station	DN	Node station	D	Station
Master station	DMT	Node station	D	Station

Table 4- 3 Abbreviations/acronyms:

	Network type / mode
D	Dedicated line / polling
DT	Dedicated line / polling with time slots
DMT	Dedicated line / Multi-master polling with time slots
DN	Dial-up network / spontaneous mode
SPN	Spontaneous network / spontaneous mode

4.1.4 Properties and restrictions of PG routing

When using PG routing with the SINAUT Diagnostics and Service tool or the SIMATIC Manager, certain special features and restrictions to the functions must be kept in mind.

Functions of the SINAUT Diagnostics and Service tool with PG routing

- *TIM Message Monitor* function
  Activating the *TIM message monitor* on a remote TIM using PG routing is not possible.

- *Firmware update* function
  When using the *Firmware update* function, remember that large amounts of data are transferred. With remote modules, long processing times of several minutes can occur.

- *Repair* function
  The *repair* function must not be used over PG routing.
  If you use the *repair* function, the flash disk of the TIM is formatted and the software completely deleted on the TIM. Following this, the module is no longer accessible over the SINAUT network. Reloading the TIM software is then only possible locally over the MPI bus.

Functions of the SIMATIC Manager with PG routing

- Display *Accessible nodes* function
  The display *Accessible nodes* function is available only for subscribers connected to the local MPI bus. This restriction applies to SIMATIC PG routing and therefore also to SINAUT PG routing.

- *Hardware diagnostics* function
  The hardware diagnostics function is available only for subscribers connected to the local MPI bus. This restriction applies to SIMATIC PG routing and therefore also to SINAUT PG routing.

- *Download* function
  The *download* of an entire station (CPU 300 plus TIM) leads to a connection abort. Since the TIM system data are downloaded first followed by a restart on the TIM, the second step of the CPU data download is interrupted.
  It is possible to repeat the CPU download. We, nevertheless, recommend that you download the block folder of the CPU and the TIM module separately.

- *Upload to PG* function
  The *upload to PG* function is available only for subscribers connected to the local MPI bus. This restriction applies to SIMATIC PG routing and therefore also to SINAUT PG routing.

Restrictions for PG routing with SINAUT ST1 components

For more information on SINAUT ST1, refer to release 05/2007 of the manual.

4.2 System requirements for PG routing

Introduction

To use the PG routing function in the SINAUT telecontrol network with SINAUT ST7, the following requirements must be met or preparations made.

- **STEP 7** with at least Version 5.1 and service pack 3 is required on the PG/PC.
- The **SINAUT software package** version V3.0 or higher must be installed on your PG/PC.
- The **RMOS operating system** of the TIM modules of the type TIM 3 and TIM 4 in your telecontrol network should be at least version 2.14. TIM modules of version 2.04 can be used with certain restrictions.
- The **TIM firmware** for the TIM 3 and TIM 4 must have a certain minimum version.
  With dedicated lines, firmware version V 3.14 is required.
  In dial-up networks, a minimum firmware version of V 3.46 is required.
- For SINAUT dedicated line or dial-up networks, a **maximum message length** of 240 must be set.
- On slow connections with only 1200 bauds transmission speed, the **Retry factor** must be set higher than 0.
- For PG routing, **new SDBs** must be transferred to the TIM modules. These are compiled with the SINAUT software package as of V3.0.

Note

PG routing is released for the Ethernet TIMs regardless of the firmware version.

4.2.1 STEP 7

STEP 7 at least version 5.1,service pack 3 must be installed on your PG/PC. Only then is your PG/PC capable of routing and it is guaranteed that the routing SDBs are compiled with the correct content.

4.2.2 The SINAUT software package

The routing SDB (SDB type 3002) can only be compiled for the TIM modules with the SINAUT software package as of the version V 3.0. The LAN SDB (SDB type 3201) also includes extra parameters required for PG routing. PG routing can only be used over TIM modules on which these routing and LAN SDBs are loaded.

Note

The routing-SDB (SDB type 3002) is compiled only for TIM modules that actually require this information.
No routing SDB is compiled for a TIM installed as a CP in an S7-300 station if there is no local network such as MPI bus or PROFIBUS DP in this station. The absence of the routing SDB in the SDB directory is not, in this case, a compilation error. Although the routing SDB is missing, this TIM and the connected S7-300 CPU can be accessed using PG routing.

Table 4- 4 Overview of the SDBs

SDB type	Contents
0	Standard SIMATIC SDB
700	Communication function block connection parameter
3002	Routing SDB
3201	LAN data
3202	WAN data
3203	SINAUT subscriber data
3205	Parameter for X connection

Apart from SDB type 0, the SDBs of the type 3201, 3202, 3203 and 3205 are always compiled for the TIM, the SDB types 700 and 3002, on the other hand, only when necessary.

4.2.3 RMOS operating system of the TIM 4

Whether a TIM of the type TIM 4 is suitable for PG routing, depends, among other things, on the version of the RMOS operating system of the TIM. In this respect, there are versions ideally suited for PG routing and versions that mean certain restrictions:

- TIM modules with an RMOS version older than 2.04 are not routing-compliant.
- TIM modules with an RMOS version as of 2.04 are routing-compliant but involve restrictions regarding the number of remote subnets. No more than 10 remote subnets may be configured in the SINAUT project.

- TIM modules with an RMOS version as of 2.08 are routing-compliant but involve restrictions regarding the number of remote subnets. No more than 20 remote subnets may be configured in the SINAUT project.
- TIM modules with an RMOS version as of 2.14 are fully routing-compliant.

The operating system of the TIM modules can be upgraded to the current version. A special cable is required for this. This can be borrowed from the hotline. Along with the cable, the current version of the operating system is supplied on diskette with instructions on upgrading using a PG/PC.

4.2.4 TIM firmware for TIM 4

Not only the version of the operating system but also the version of the TIM firmware for the TIM 4 decides whether a TIM is suitable for PG routing. In this respect, there are versions ideally suited for PG routing and earlier versions that mean certain restrictions.

For PG routing, a TIM firmware version of V 3.46 is generally recommended. It depends on the driver activated on the TIM (dedicated line or dial-up network driver) whether an older version will be adequate. PG routing over the dedicated line driver is possible as of firmware version V 3.14. If the dial-up network driver is activated on a TIM module, firmware version V 3.46 or higher must be installed.

4.2.5 Settings for SINAUT networks

Retry factor and maximum message length

PG routing can only be used when the *maximum message length* network parameter for the SINAUT networks is set to a maximum of 240. Since this value is normally entered as default in the *Properties* dialog for SINAUT networks, this is normally already set.

If a transmission speed of 1200 bauds is set in a SINAUT network, make sure that the retry factor is higher than 0. We recommend that you use the default values available for the retry factor in the configuration dialogs:

- For dedicated lines: 3
- For dial-up networks: 7

4.2.6 Recompiling system blocks

If you want to enable PG routing in an existing SINAUT ST7 system, the project must be recompiled in the subscriber management of the SINAUT configuration tool (as of V 3.0). It is adequate to simply recompile the new SDBs there. The programs for CPU modules do not change for PG routing.

After compiling the SDBs, the entire SINAUT system configuration must be downloaded and activated on the TIM modules.

4.2.7 Downloading newly compiled SDBs to TIM modules

If new SDBs have been compiled for an existing SINAUT ST7 system, they must then be downloaded to the relevant TIM modules of the system.

Without PG routing, SDBs are downloaded locally to each TIM of the MPI bus. A further option for downloading SDBs from the master station to the remote TIM modules is described in the section *Central SDB downloading with PG routing*.

Note

The new system data blocks are downloaded to a TIM module using the SINAUT Diagnostics and Service tool function *Download SDBs*. You will find more detailed information there.

To activate newly downloaded SDBs on a TIM module, the TIM must be restarted. The consequences of restarting a TIM module are, however, as follows:

- The connection from the PG to the TIM is terminated.
- The connections from the TIM to other SINAUT partners (SINAUT connections) are terminated. This leads to error messages on the partners of the TIM module.
- In the case of a node TIM, the connections to the downstream stations are also reported as being disrupted.
- With a node TIM, any data messages stored on the TIM are lost during the restart. This is particularly relevant in station TIMs in the dial-up network.

When downloading SDBs to TIM modules, you should therefore note the following points:

- Before you transfer the SDBs, you should give the TIM the opportunity of transferring any messages stored on it.
- After restarting the TIM, the SINAUT connections are automatically re-established. If the connection from the PG to the TIM is required, however, it must be activated from the PG of the user.

4.2.8 Central SDB download using PG routing

SDBs can be downloaded as previously for each TIM locally over the MPI bus. It is, however, conceivable to perform this from the master station; in other words, to use the options of PG routing although the system was not activated for this function. This method can be considered if you already know that the TIM modules in the stations are suitable or prepared for PG routing in terms of the operating system and TIM firmware.

In the following sample configuration, the central downloading and activation of the SDBs is performed in four consecutive steps:

Step 1:

First download the newly compiled SDBs to the TIM module in the master station (in the illustrated example, this is TIM 4 a) and then activate the SDBs by restarting the TIM module. TIM 4 is now routing-compliant.

Repeat this procedure for all TIM modules in the master station if there is more than one TIM module.

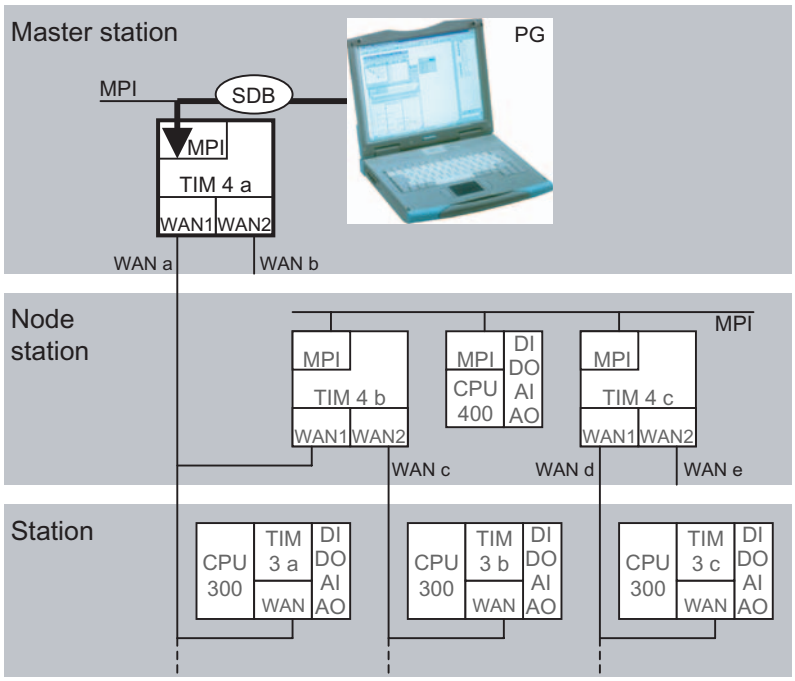


Figure 4-5 Step 1 of central SDB download in the sample configuration

Over the now routing-compliant master TIM 4, the PG now has access to all station or node TIM modules connected directly to this master TIM over WAN a and WAN b. In the example, the PG can reach both TIM 4 b and TIM 3 a TIM modules over WAN but not the local CPU modules, the TIM 4 c of the node station.

Since the two TIM modules TIM 4 b and TIM 3 a are now accessible, the PG can run the diagnostics and service functions for both these TIM modules. This means that SDBs can also be transferred to these two TIM modules by the PG.

Step 2:

Download the newly compiled SDBs to the TIM modules TIM 4 b and TIM 3 a and then activate the SDBs by running a restart on the two TIM modules. TIM 4 b and TIM 3 a are now routing-compliant.

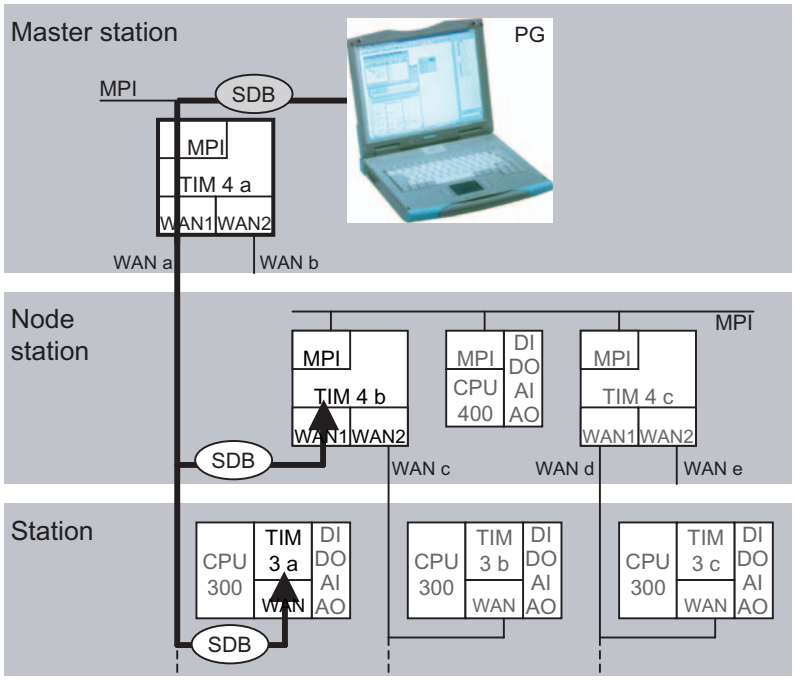


Figure 4-6 Step 2 of central SDB download in the sample configuration

Over the two routing-compliant modules TIM 4 b and TIM 3 a, the PG can now access all locally connected CPU and TIM modules. All CPU and TIM modules connected to a routing-compliant TIM over the backplane bus (with S7-300), over MPI, or over WAN are counted as local.

In the example, the PG can now reach the following CPU and TIM modules:

- In the stations, the CPUs connected to TIM 3 a over the backplane bus.
- In the node station, the CPU and the TIM 4 c over MPI, the station TIM 3 b connected to TIM 4 b over WAN c, but not the local CPU there.

The reachable CPU modules can now be remotely programmed. The PG can once again perform all the diagnostics and service functions for the two newly reachable TIM modules TIM 4 c and TIM 3 b. The PG can therefore also transfer SDBs to these TIM modules.

Step 3:

Download the newly compiled SDBs to the TIM modules TIM 4 b and TIM 3 a and then activate the SDBs by running a restart on the two TIM modules. TIM 4 c and TIM 3 b are now routing-compliant.

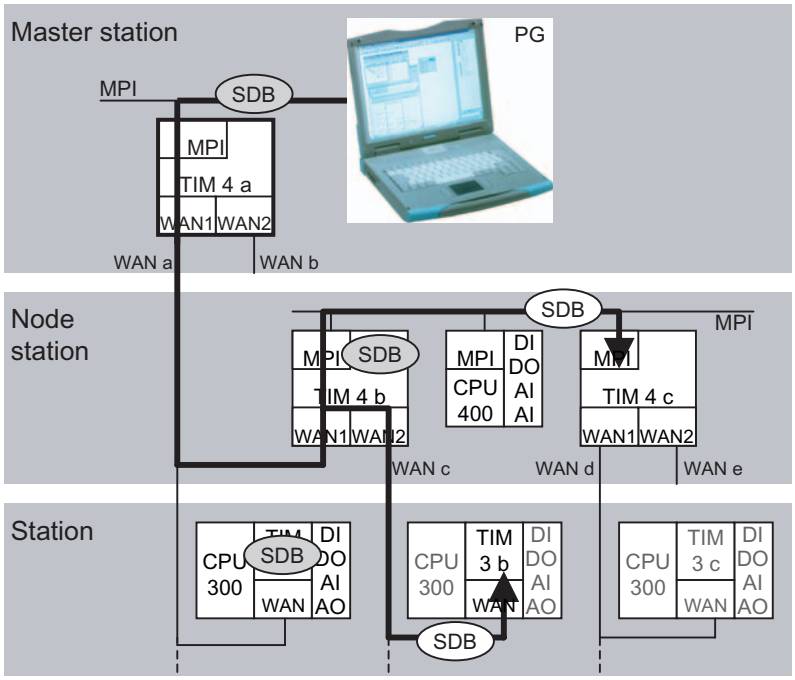


Figure 4-7 Step 3 of central SDB download in the sample configuration

Over the two routing-compliant modules TIM 4 c and TIM 3 b, the PG can now access all locally connected CPU and TIM modules. In the example, the PG can now reach the following CPU and TIM modules:

- In the stations, the CPUs connected to TIM 3 b over the backplane bus.
- The station TIM 3 c, connected over WAN d to the TIM 4 c in the node station, but the CPU connected locally to TIM 3 c in the station not yet.

The reachable CPU module can now be remotely programmed.

The PG can once again perform all the diagnostics and service functions for the newly reachable TIM module. The PG can therefore also transfer SDBs to this TIM module.

Step 4:

Download the newly compiled SDBs to the TIM module TIM 3 c and then activate the SDBs by running a restart on the TIM module.

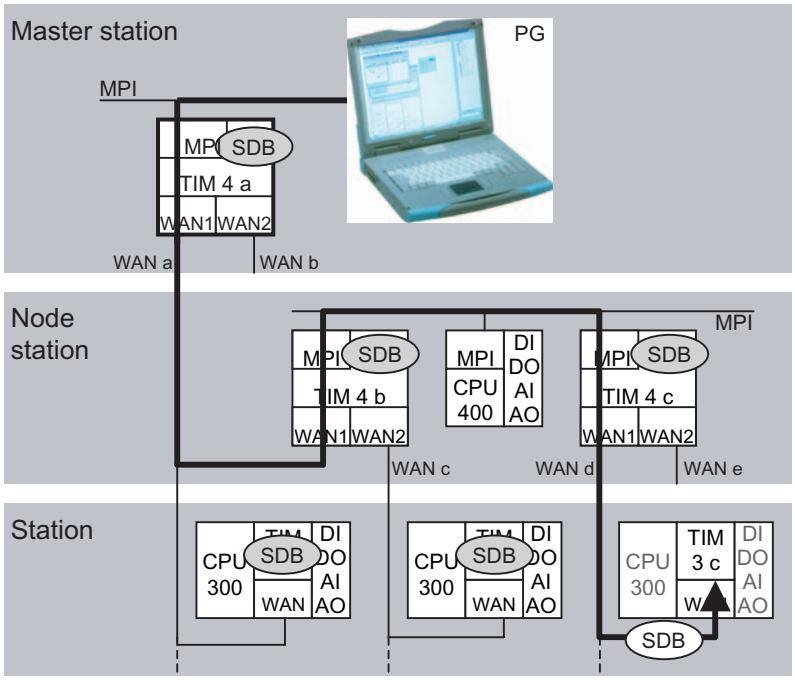


Figure 4-8 Step 4 of central SDB download in the sample configuration

TIM 3 c is now also routing-compliant. The CPU module attached to TIM 3 b can now be remotely programmed.

4.3 Application of PG routing

Before you perform PG routing over the SINAUT telecontrol network with your PG of PC, you must first adapt the properties of the PG/PC interface and set the assignment of the PG/PC in the SINAUT network.

4.3.1 Properties of the PG/PC interface

Adapting the PG/PC interface

1. Open the *Control Panel* window by clicking on the *Start / Settings / Control Panel* menu.
2. Select the *Set PG/PC interface* icon.
3. In the *Set PG/PC Interface* dialog, set the MPI interface in the *Interface parameter assignment used* field.
4. Then click on the *Properties* button.

5. Confirm the warning dialog with *Yes*. The *Properties* dialog opens.
6. In the *MPI* tab of the *Properties* dialog, select the option *100 s* in the *Timeout* list.
7. Close the *Properties* and *Set PG/PC Interface* dialogs with the *OK* button and then close the *Control Panel* window.

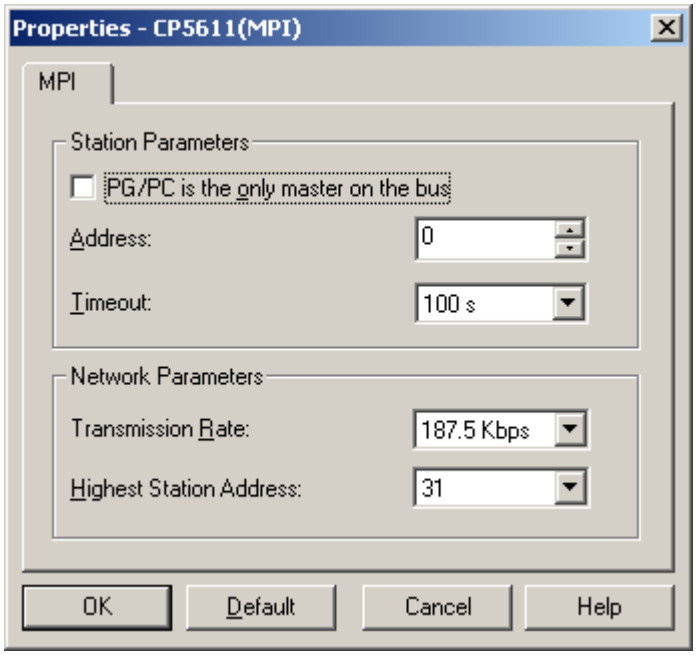


Figure 4-9 *Properties* dialog of the interface in the Control Panel

This completes the adaptation of the PG/PC interface.

4.3.2 PG/PC assignment in the SINAUT network

Before you can use PG routing with a PG/PC over SINAUT networks, this must be configured and assigned within a SINAUT project.

Assigning the PG/PC

The PG/PC is assigned in the network using the SIMATIC network configuration tool NetPro.

1. Open the project in which you want to use PG routing in the SIMATIC STEP 7 *NetPro* network configuration tool.
2. Open the NetPro catalog if it is not already open.
3. Drag a PG/PC to the network window from the NetPro catalog directory *Stations* and place it at a suitable position.
4. Right-click on the PG/PC you have just installed. A context menu opens.
5. In the context menu, click on the *Assign PG/PC* option. The *Properties* dialog opens.

6. Select the MPI network to which the configured PG/PC is connected in the *Configured interface* field of the *Properties* dialog.

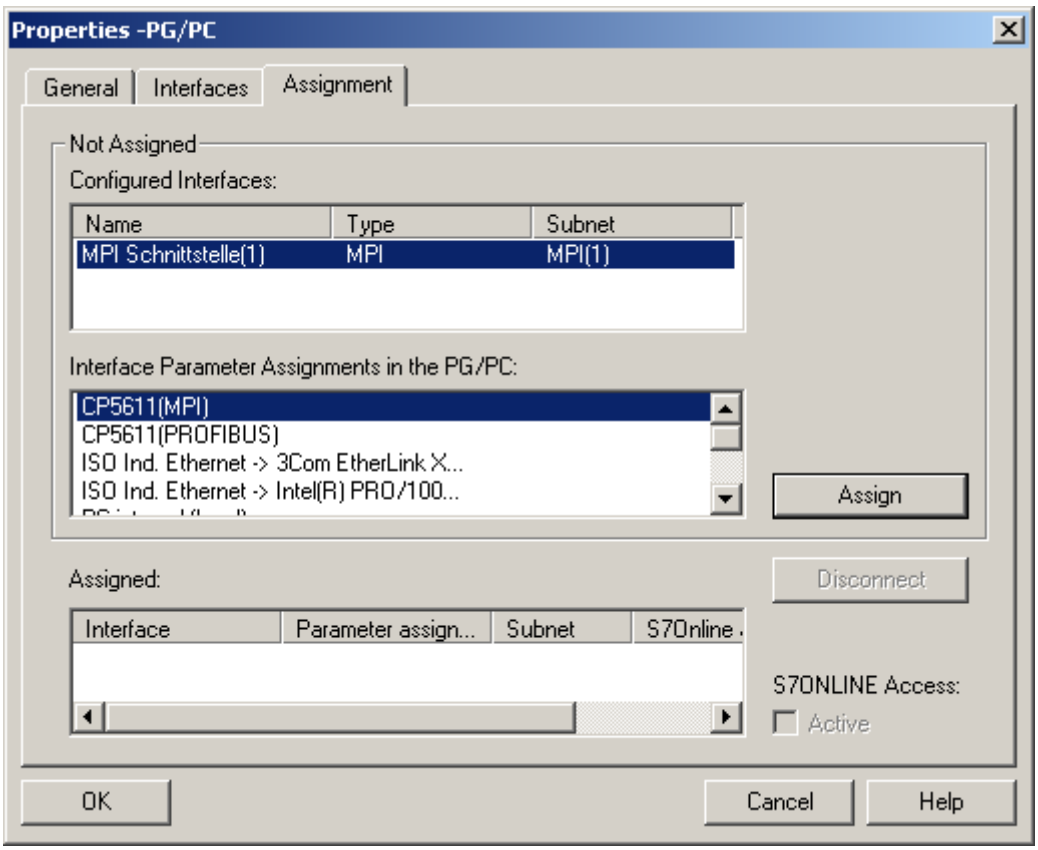


Figure 4-10 *Properties - PG/PG* dialog / *Assignment* tab in NetPro

7. Select the MPI interface you want to use in the *Interface Parameter Assignments in the PG/PC*.
8. Click on *Assign*. The assigned MPI interface is displayed in the *Assigned* box. The interface is now enabled for PG routing access.
9. Close the dialog with the *OK* button. The successful assignment of the PG/PC is indicated by an MPI connection on a yellow background and a yellow arrow pointing upwards in the PG/PC icon in *NetPro* and in the *SIMATIC Manager*.

10. Save your project in *NetPro*.
11. Connect your PG/PC to the MPI bus to which you assigned your PG/PC in *NetPro* over a PG cable.

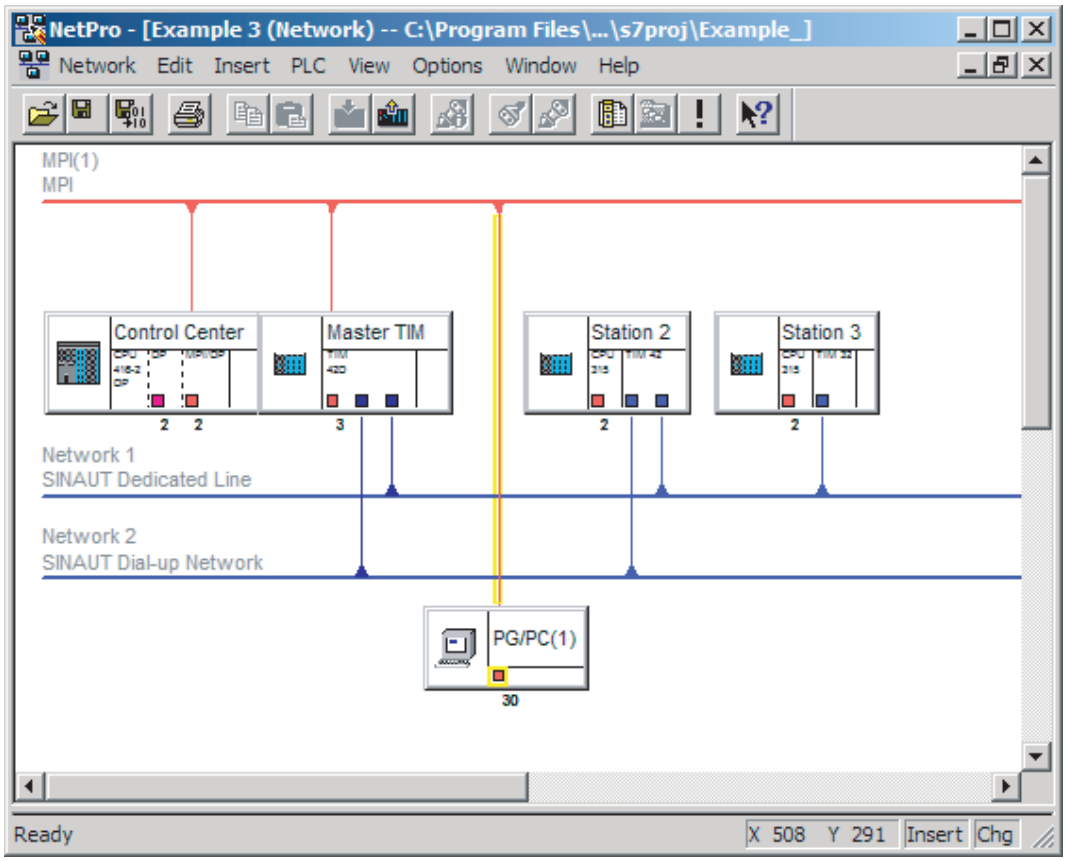


Figure 4-11 Project view in *NetPro* with assigned PG/PC

Note

As long as you leave your PG/PC connected to the point in the network as you assigned it in *NetPro*, you do not need to cancel the assignment. Not even if you want to turn off the PG/PC. Each time you turn on the PG/PC and open the project, you can use PG routing again immediately. You do not need to make settings or assignments again.

If, on the other hand, your PG/PC is not always at the same location or if you change to different projects on your PG/PC, we strongly recommend that you always cancel the assignment before your PG/PC is turned off or before you change to a different project. This ensures that the PG/PC can be assigned again when used at a different location or in a different project.

4.3.3 PG routing in dial-up networks

There are PG functions that maintain constant message traffic with the CPU or the TIM after they have started because these functions want to update themselves constantly. Such a function in a dial-up network would lead to a permanent connection with the relevant station until you deselect the function.

Other PG functions execute only once. They do not need constant updating. In a dial-up network, in this case, the dial-up connection is established briefly and then terminated again.

If a continuous connection to a dial-up station is required with PG routing, it is advisable to start a PG function at the beginning that requires constant updating regardless of the actual PG function that has just been activated. This function should then be maintained while PG routing is being used thus forcing a permanent connection. At the same time, other PG functions can be performed.

The PG function most suitable for maintaining a permanent connection is the *Operating mode* function. This requires little time to execute and the load on the other data traffic caused by the constant updating is only minimal. Other suitable functions include *Module Information* and *Set Time of Day*.

PG routing places load on the transfer of process data; in other words, this data is transferred to the control center while the PG connection is established. This will will be slower than when no PG routing is activated.

On the other hand, the parallel transmission of data messages also slows down the reactions to PG functions that have started. This is particularly the case when the station TIM is currently being used for PG routing and this TIM has a lot of stored messages, PG routing will be very slow at the beginning. It is therefore advisable to give the station TIM the opportunity to transfer its messages before intensive PG functions are started.

4.3.4 Canceling the PG/PC attachment in the SINAUT network

Canceling the PG/PC assignment

If a PG was assigned in a project and then needs to be used at a different location, for example locally connected directly to a CPU or TIM, the assignment must first be canceled. Follow the steps outlined below:

1. Right-click on the PG/PC that is still assigned in the project you have opened in *NetPro*.
2. In the context menu that opens, select *Cancel PG/PC Assignment*.
3. Acknowledge the warning dialog with *OK*.
   The PG/PC assignment is now canceled, this is indicated as follows in the network image of *NetPro*:
   - The connecting line from the PG/PC to the MPI bus is no longer on a yellow background.
   - The yellow arrow in the PG/PC icon disappears
4. Save your project in *NetPro*.

Glossary

Alarm logging (runtime)

Alarm Logging controls the acquisition and archiving of events and provides display and operator input options. Using the message blocks, message class and message type structure elements, the configuration engineer can class the events according to their significance and allow the operator a fast evaluation of the status of the system.

Analog value

An analog value is an analog process variable such as pressure, temperature etc. It is acquired over an analog input as a current or voltage value and converted by this module to a binary-coded value. In total, the converted value occupies 1 word; in other words, 16 bits including sign bit.

APN - Access Point Name

DNS host name of the access point for an external network (in this case: access point in the GPRS network to the Internet).

Automation program

The automation program is the program section on the CPU that monitors and controls the technological process.

COM port

→ *RS-232*

Command

A command is binary information that is transferred once as a 1 when there is a signal change from 0 to 1. The trailing edge from 1 to 0 does not trigger transmission again. At the receiving end, a command is either output as a pulse (selectable duration) or it is reset by the local user program after it is has been executed. 8 commands are put together to form a byte.

When inputting and outputting commands, reliability and safety are important. At the input end, for example, there is a check to determine whether only one command is pending at the time of acquisition (1-out-of-n check). Only then is the transmission triggered. If several commands are pending at the same time, an error is detected. There is no transmission. To transmit a command byte, a total of 1 word is used: One "original" command byte and a copy. At the receiving end, a command is only output when the "original" command byte and the copy have the identical content and when only 1 command was received (once again a 1-out-of-n check).

Conditional spontaneous message

In the dial-up network, you can specify whether or not a change causes a "conditional spontaneous" or "unconditional spontaneous" transmission for each individual message.

Conditional spontaneous messages are initially only entered in the send buffer of the TIM. They are only transmitted when a connection is established to the partner for whatever reason, for example because an unconditional spontaneous message needs to be transmitted or because the partner calls.

Even when using pay by volume transmission in a GPRS network, message prioritization "conditional spontaneous" can also be used. Such a message is not transmitted immediately, but is first buffered. In a GPRS network, the TIM stores "conditional spontaneous" messages in the following situations:

- When the collected messages reach or exceed a size of 202 bytes.
- When an important message (priority "conditional spontaneous" or "alarm") needs to be transmitted immediately.
- When the collected messages have not yet reached a volume of 202 bytes, but the TCP/IP keepalive interval expires.

Configuration

During configuration, communication- and connection-specific system settings are made for each device.

Counted value

A counted value (for example amount of flow) is acquired over a digital input as a pulse train and totaled to produce a binary-coded value. A counted value is 2 words: 28 bits for the binary-coded value 4 display bits.

CP

Communications processor

CPU

(Central Processing Unit)

The CPU handles the central sequential control and coordination of all activities of the module.

CSD

(Circuit Switched Data)

Service in GSM for wireless transmission of data at 9 600 bps full duplex. Connections can be established to other GSM devices, to analog modems or to ISDN modems in the fixed network. The connection establishment can be started at both ends. Only dial-up connections are supported.

Data manager

The accrued data is managed by the data manager in WinCC. The data manager works with data created in the WinCC project and data stored in the project database. It handles the entire management of WinCC tags while WinCC is in runtime mode. All WinCC applications request the data from the data manager in the form of WinCC tags. These applications include Graphics Runtime, Alarm Logging Runtime and Tag Logging Runtime.

To allow WinCC to communicate with the widest variety of data sources (programmable controllers, ST7cc servers etc.), various communications drivers are used.

A communications driver is a C++ DLL that communicates with the data manager over a specified interface known as the channel API. The WinCC tags are supplied with process values over the communications driver.

Data messages

The actual transmission of data takes the form of data messages. These contain a fixed amount of a specific information type.
There are status messages, analog value messages, counted value messages, command messages, setpoint messages, parameter messages and data messages for the various information types.

DCF77 radio clock

DCF77 is a time signal transmitted on the normal frequency 77.5 kHz as encoded time information. Reception of the time signal is restricted to Western Europe.

Some TIM variants (TIM 4VD, TIM 42D, TIM 43D, TIM 44D, TIM 4RD) are equipped with a DCF77 receiver module that can receive the DCF77 time signal either over an indoor or outdoor antenna. The ST7 time management currently requires the existence of a DCF77 time signal to allow time synchronization of the stations and the control center throughout a network.

For applications that cannot receive the DCF77 time signal, SINAUT provides a GPS receiver module with a GPS outdoor antenna that converts the GPS Time signal for the DCF77 receiver module.

Direct data communication

Communication between two stations. In dial-up networks or in Ethernet, this is achieved by direct connection establishment between the two stations. On dedicated lines and in wireless networks or in GPRS mobile wireless service or when using the MSC protocol, direct communication is more "indirect". It is switched via the closest → master TIM.

DNS

Domain Name System, a distributed database that manages the name space in the Internet

DSL

Digital Subscriber Line

DSL is used mainly for transmission of telephone and Internet data. By using a higher bandwidth, data can be transmitted at transmission speeds in the two to three figure Mbps range.

Ethernet / Industrial Ethernet

Industrial Ethernet is a powerful communication network complying with the international standard IEEE802.3 (Ethernet) that was optimized to meet the requirements of industrial application. Ethernet is designed with a linear or star topology. The transmission media are shielded coaxial cables, twisted pair, or fiber-optic cables. SIMATIC NET Industrial Ethernet uses both the Ethernet and Fast Ethernet standards.

Ethernet TIM

→ *TIM*

Firewall

A firewall is a network component via which a secure network can be linked with an unsecure network. The task of a firewall is to control data exchange between the networks.

General request (GR)

With a general request, subscribers in a SINAUT network can request a current process image from their communications partners. This happens automatically when a disrupted connection has been restored or when a failed partner reports a restart. Apart from the automatic general request, a general request can also be triggered at any time by the user program or from the control center.

TD7onTIM does not support the general request.

Global Script (Runtime)

The term Global Script means all the C functions and actions that can be used throughout the project or over several projects. C actions are used in process control during runtime.

GPRS

(General Packet Radio Service)

GPRS is an expansion of GSM mobile wireless that adds packet-oriented data transmission. Network connections are established over GPRS either in the Internet or in private networks. The data is transmitted using the Internet protocols TCP/IP or UDP/IP.

GPS

(Global Positioning System)

GPS is a worldwide US satellite navigation system for highly accurate location, navigation and time distribution. It operates with 24 orbiting satellites (21 operational and 3 spare satellites) on six satellite orbits at a height of approximately 20,000 kilometers. Each satellite contains an atomic clock whose time is transmitted continuously along with the orbit data.

The GPS receiver receives data from a maximum of six satellites and calculates its position based on these values. Once the position has been calculated, the transmission time of the data from the individual satellites can be calculated. The GPS world time (UTC) is calculated in the system based on these values.

Graphics Designer

The Graphics Designer is a vector-oriented drawing program for creating process pictures. Complex process pictures can be created using a wide variety of graphic objects from an object and style palette.

ST7cc Config uses the ODK interface of the Graphics Designer to create picture typicals for the SINAUT subscribers and to make these available to the WinCC configuration engineer in a sample picture for use in further process pictures.

GSM

(Global System for Mobile communications)

Worldwide standard for wireless transmission of voice, data, fax and text messages (SMS). GSM-based wireless networks can be found in many countries throughout the world. A distinction is made according to frequencies: PCS 850 MHz, GSM 900 MHz, DCS 1800 MHz and PCS 1900 MHz.

Image memory / send buffer

A TIM has a send buffer and an image memory for buffering send messages.

Before it is forwarded to the communications partner, each message is entered in the send buffer. Once it has been sent, the message is cleared from the send buffer. A message is transmitted according to either the → send buffer principle or the → image memory principle. The user specifies the transmission principle during configuration of the data messages.

The storage capacity of the send buffer depends on the particular TIM type and ranges from 10 000 to 56 000 data messages.

Image memory principle

A fixed position is reserved in the image memory for each data message transferred to the TIM for transmission. Each newly transferred message always overwrites the old message in the image memory. The image memory therefore contains all data messages with their most recent content from the process.

If a send message is entered using the image memory principle, only a reference to the location of the message in the message image memory is entered. If the TIM has not yet been able to transmit the message when the same message is transferred to it again, the message is not entered in the send buffer a second time, but rather the image is simply updated.

At the time of transmission, the message is sent with its latest content from the image memory. Only then can the message be entered in the send buffer again.

Transmission using the image memory principle achieves the following:

- There is less load on transmission link, fewer messages are transmitted.
- There is less load on the send buffer of the TIM; an image memory message is entered a maximum of once in the send buffer.

See also → Image memory / send buffer

IP

Internet Protocol

IP address

An IP address consists of a numeric code made up of four numbers from 0 to 255 (for example 192.168.0.55). It is the numeric address of a particular computer in the network / Internet.

ISDN

(Integrated Services Digital Network)

ISDN integrates various services in one transmission network. ISDN networks integrate telephone, telefax, teletext, Datex-J, video phones and data transfer. This makes a wide variety of digital services available to the user: Language, texts, graphics and other data.

LAN

(Local Area Network)

Local area network.

Local data traffic is possible in the SINAUT ST7 system via the LAN types "MPI" and "Industrial Ethernet".

In SINAUT, the LAN is also used as a "WAN", for example when communication via the Internet is required or when GPRS stations transfer their data onto the Internet to a master station via a GPRS network.

Local buffer

If the ST7cc server cannot forward its data to WinCC, all messages (ST7 data messages and organizational messages) are stored in the local buffer. Once WinCC becomes available again, the buffered messages are processed. This mechanism achieves two aims:

- Form the perspective of the stations, the master station can be reached when WinCC is not available.
- General requests resulting from temporary deactivation of WinCC are avoided.

Local TIM

A TIM connected to a PC (ST7cc, ST7sc) or an S7 CPU over the MPI bus, Industrial Ethernet or an IP-based network.

LTOP

(Line Transformer with Overvoltage Protection)

Copper dedicated lines are highly susceptible to electromagnetic interference. The coupling of extraneous voltages can be inductive or capacitive, for example due to the effects of lightning. Direct conductive coupling is also possible due to bad insulation.
The LTOP overvoltage protection modules limit extraneous voltage and overvoltage to a non-critical level. The floating transformer also provides electrical isolation preventing coupling of voltages into other cable sections. An LTOP protects persons and investment and is therefore an indispensable safety element in private dedicated line networks.

Main and subcycle

The sequence of the polling cycle can be structured on the master TIM by assigning individual polling stations to a main cycle or a subcycle. The subcycle is always activated at the end of the main cycle; in other words, once all stations from the main cycle have been polled, a certain, selectable number of stations are polled in the subcycle. Following this, all the stations in the main cycle are polled again.

Master TIM

WAN interface of a TIM set to the "master" function. Use in control centers and node stations.

MPI

The multipoint interface (MPI) is the programming device interface of SIMATIC S7. Devices with an MPI interface (for example a TIM), can also communicate with each other (MPI bus).

MSC protocol

The MSC protocol is a proprietary protocol and is used in SINAUT ST7 for the data exchange between TCP/IP-compliant devices in Ethernet, landline or mobile wireless networks. The MSC protocol provides an authentication mechanism and simple encryption of data. To transfer messages in IP-based WANs, the MSC protocol is linked into the protocol of SINAUT ST7.

Multimaster polling with time slots

When stations need to communicate with more than one master station in dedicated line or wireless operation, the multimaster polling with time slots mode is used. Each of the connected master stations is assigned one or more defined time slots per minute for polling. The master stations then have their turn to poll in every minute.

Node station

A node station is a station located between the master station and stations in the hierarchy of a telecontrol network. One or more subordinate stations are connected to a node station. The data traffic between these stations and the master station is handled via the node station. Direct data exchange between the node station and the subordinate stations is also

possible. It is possible to have several node stations in a SINAUT network; in other words the data traffic between stations and master station can involve two or more node stations.

Node station TIM

WAN interface of a TIM in a node station set to the "node station" function.

If the TIM used in a node station has two WAN interfaces, the "station", "node station" and "master station" functions can be used in any combination on one TIM.

OP

Operator panel, HMI device

OPC

(OLE for Process Control)

OPC includes a series of specifications for data exchange in automation engineering between controllers, alarm transmitters etc. and control systems.

Organizational message

Organizational messages are used to execute organizational system functions.

These include, for example:

- General requests
- Time synchronization
- Counted value storage
- Coordinated connection establishment and termination in a dial-up network
- Message indicating station startup and station failure
- Requests for and transmission of subscriber records

Parameter assignment

A device is assigned the parameters it requires for starting up when it is supplied with data.

Permanent call

A permanent call does not interrupt the normal polling cycle; it is always executed alternating with the standard poll from the normal polling cycle.

PG / programming device

A PG is a personal computer designed specifically for use in industry. A PG is fully equipped for programming SIMATIC automation systems.

PG Routing

Using PG routing, it is possible to access programmable modules or modules with diagnostic capability beyond network boundaries from a programming device (PG) or computer (PC).

PLC

Programmable logic controller

Polling

Polling is a method of SINAUT data transmission in which a → master TIM controls the data exchange in the telecontrol network.

Using a polling message, the master TIM instructs the connected stations (station TIMs) one after the other to transmit their stored data messages to the master TIM. If a polled station has no stored data messages, it responds with an acknowledgment message and the polling cycle then continues by polling the next station.

A station that has stored data sends a single data message or, if block transfer was configured, several messages in a block. In the reply, the station indicates whether it still has other stored data messages. In this is the case, the station is then polled again immediately until all stored messages have been transmitted or until a selected maximum number of messages have been sent. The polling cycle then moves on to the next station.

Polling mode

→ *Polling*

Polling with time slots

The polling with time slots mode is used in a wireless network in which the use of the radio frequency assigned by the registration authorities must be shared with other users. Each user typically has 6 seconds per minute to exchange data with its stations. The frequency must then be released for other operators. During the allocated time slot, this pooling variant functions like a normal polling system.

PROFIBUS

PROFIBUS is the open, internationally standardized (EN50170) bus system for process and field communication with field devices and for data communication within an automation cell. The uses of PROFIBUS range from production and process automation to building automation.

Protocol

A protocol is a set of rules for controlled transfer of data. Protocols, for example, specify the data structure, the structure of data packets and the coding. Protocols can also specify a control mechanisms and hardware and software requirements.

Remote buffer

The remote buffer is set up only for redundant ST7cc. The ST7cc server recognizes whether or not the redundant mode is required based on the existence of the redundancy license.

The remote buffer is organized as a ring buffer and records all incoming messages so that it can be used as a data source for the redundant partner during a restart. If the partner of a redundant ST7cc system starts up again, it can recognize the time for which messages are missing and can request these from the redundancy partner. The remote buffer is necessary to ensure data consistency when using a redundant ST7cc system.

Requested message

Polled messages are data messages of a station or node TIM with a special identifier indicating that they were sent in response to a general request from the master station.

RJ-12

This describes a 6-pin connecting cable with a standardized modular (Western) connector.

RJ-45

This describes a 8-pin connecting cable with a standardized modular (Western) connector.

RS-232

RS-232 is a standard for serial (i.e. bit-by bit) data transmission with +12 V and -12 V signals. RS-232 is a Recommended Standard of the Electronic Industries Association. For the RS-232 interface, 9-pin and 25-pin connections with D-sub connectors are normal. These are sub-miniature connectors with a D-shaped face.

RS-485

RS-485 is a standard for data transmission with 5 V differential signals. The RS-485 interface uses only one pair of wires and is operated in half duplex. The connection is multipoint-compliant; in other words, up to 32 subscribers can be connected.

Send buffer

→ *Image memory / send buffer*

Send buffer principle

If a send message is transmitted using the send buffer principle, each time the message is transferred to the TIM, it is entered completely in the send buffer. If such a message cannot (or should not) be transmitted immediately, it may exist more than once in the send buffer. When it is sent, the message is taken completely from the send buffer and transmitted.

See also → Image memory / send buffer

Setpoint

A setpoint is a selected digital or analog value that is transmitted once after the value has been set. The entered value is recalculated when necessary. A setpoint is always transmitted as 1 word. At the receiving end, the setpoint can either the output directly to the process as an analog signal (for example to an external controller) or the value is made available to the local program for further processing (setpoint for internal controller, limit value, threshold value etc.).

Setpoint and command input are interlocked for safety reasons; in other words, a setpoint input cannot be made at the same time as a command input. In this case, the acquisition program recognizes an error. Neither the setpoint nor the command are transmitted.

SIM card

(Subscriber Identity Module)

The SIM card is an identification card for a subscriber of a GSM mobile wireless network.

SIMATIC S7

Siemens automation system

Simple Internet communication

In SINAUT ST7, simple Internet communication means data exchange between TCP/IP-compliant devices in Ethernet, landline or mobile wireless networks using the → MSC protocol.

SINAUT

(SIemens Network AUTomation)

Station control system or telecontrol system based on SIMATIC S7

SINAUT message

An ST7 message contains the data of an ST7 object for transmission. Depending on the object type, a message can contain either all data of an ST7 object or only a contiguous subarea of the object data.

SINAUT object

A SINAUT object contains the data of one or more process variables such as analog values, commands, calculated values, status information on motors, sliders etc. An ST7 object has type-specific processing functions and change checks assigned to it to minimize the communication traffic in the WAN. Type-specific processing functions include, for example, threshold checks or mean value calculation with the object type for analog values. The change check is designed so that a message is generated only when the object data has changed compared with the last time its value was transferred or when the type-specific processing enables generation of a message because the object data is "worth" transferring.

SINAUT ST7 protocol

This protocol is used in the SINAUT ST7 system for data transmission via classic WAN networks and IP-based networks. It is a further development of the SINAUT ST1 protocol.

SINAUT TD7 Library

The SINAUT TD7 library is a software package with blocks for the CPU. The package can be run both on an S7-300 or an S7-400 CPU (except the S7-400H CPUs). There are only a few blocks intended specifically for the S7-300 or S7-400 CPUs.

The SINAUT TD7 software in the stations allows change-driven transmission of process data between the individual CPUs and the control center, for example ST7cc. Failure of connections, CPUs, or the control center are displayed. Once a problem has been corrected or the CPUs or control center have started up, data is updated automatically. When necessary, data messages can be given a time stamp.

There are two variants of the SINAUT TD7 software package:

- TD7onCPU

  Library with SINAUT blocks that are linked into the program of the CPU and are executed there. The library can be used with all TIM types and is suitable for all S7-300 and S7-400 CPUs (except the S7-400H CPUs).

- TD7onTIM

  SINAUT TD7onTIM is the TD7 variant that runs on the TIM. It is available for the → Ethernet TIMs and can only be used in an S7-300 station. TD7onTIM and TD7onCPU can only be used as alternatives in a station. Simultaneous use is not possible.

SINAUT TD7onCPU

→ *SINAUT TD7 Library*

SINAUT TD7onTIM

→ *SINAUT TD7 Library*

SMS

(Short Message Service)

The short message service in the GSM standard is used to transfer short text messages to mobile wireless users.

When the short messages are transferred, they are first transferred to the SMS center (SMSC) using a store-and-forward technique. They are buffered there and then forwarded to the recipient. The sender can query the status of the message in the SMS center or can request acknowledgment of delivery.

Spontaneous message

In SINAUT networks, messages are always transmitted spontaneously; in other words, messages are created and transmitted only when changes to process values occur or event-driven. These messages are known as spontaneous messages.

See also → "Conditional spontaneous message" and → "Unconditional spontaneous message"

Spontaneous mode

The spontaneous mode is a method of SINAUT data transmission in which subscribers can exchange data among themselves directly. Here, no central entity is necessary in the form of a master TIM as in polling mode (see "Polling mode"). The spontaneous mode is intended for data transmission in dial-up networks and for communication via IP-based networks.

For transmission in a dial-up network and in pay-by-volume/time IP-based networks (for example GPRS), the data to be sent is assigned different priorities during parameter assignment ("high" or "normal", with TD7onTIM also "alarm"). When data with high or alarm priority are ready for transmission, the dial-up connection is established immediately. If the data has "normal" priority, it is first stored on the TIM. This data is then sent the next time a connection is established to the partner for whatever reason. This can, for example, be the situation when information with higher priority needs to be sent or when the partner establishes a connection to exchange data.

Squelch

Squelch

Squelch reduces the interfering noise in communications during breaks in transmission. This increases the signal-to-noise ratio.

ST7 message

ST7 messages consist of a frame, an area for addresses and control fields (message header) and an area for net data (object data) with the time stamp. The ST7 messages are divided into organizational messages and data messages. The structure of an ST7 message is based on the guidelines of the ST7 protocol.

ST7 protocol

This protocol is used in the SINAUT ST7 system for transmitting process data over WANs.

ST7cc

SINAUT ST7cc is the ideal control center system based on SIMATIC WinCC for SINAUT ST7.
It is specially designed for event-driven and time-stamped data transmission in the SINAUT system. It avoids the possible loss of data that can occur with cyclic polling in WinCC. It also ensures the use of the correct event time supplied by the SINAUT stations for all WinCC messages and archive entries. The integrated process image contains all process data as well as the status of all SINAUT subscribers in the network and makes this data available directly to WinCC for fast transfer to the process image.

ST7cc Config

ST7cc Config allows the parameter values required for message processing to be specified. When the WinCC data framework is generated (in other words, message management), the individual WinCC messages are generated and imported into WinCC.

ST7cc tag

An ST7cc tag is a data section from the data area of a SINAUT object that is managed and processed as a separate information unit in the ST7cc server. The tags are processed, however, in ST7cc and in WinCC. When the tags are defined, different processing can be assigned to them depending on their type. A tag can contain both a process value as well as status information from system components. System components are the SINAUT subscribers.

ST7cc tag management

ST7cc tag management covers all ST7cc tags. The content of the ST7cc tags represents the current process image. WinCC writes and reads the ST7cc tags.

ST7sc

The SINAUT system allows the networking of SIMATIC stations with a control center over a WAN (Wide Area Network). This control center can also be a SIMATIC station or a PC-based control center, for example, WinCC with the SINAUT ST7cc add-on.

SINAUT ST7sc allows the manufacturers of control systems to connect to SINAUT without needing to integrate a SINAUT interface. Communication is over OPC: As an OPC server, SINAUT ST7sc forms the interface between the SINAUT system and a control system connected as an OPC client.

The OPC interface is also suitable for data exchange with other applications, for example, the Microsoft Office application Excel.

Station

In the SINAUT world, the term station includes the entire hardware components required for acquisition, processing and communication with other stations or a master station. A station can, for example, consist of a modem, a TIM and a programmable controller (in turn consisting of a CPU and I/O modules). A SINAUT station can also include several programmable controllers, or, in the case of a node station, several TIMs.

Station TIM

WAN interface of a TIM set to the "station" function. Use in stations.

Status message

A status message is a process status (for example pump on, valve open) or alarm (for example limit value exceeded). This is binary information with the possible values 0 or 1. Eight status messages are put together to form one byte.

STEP 7

The basic STEP 7 software is the standard tool for the SIMATIC S7, SIMATIC C7 and SIMATIC WinCC automation systems.

Subnet mask

The subnet mask specifies which parts of an IP address are assigned to the network number. The bits in the IP address whose corresponding bits in the subnet mask have the value 1 are assigned to the network number.

TCO (TIM Connect)

The TCO component monitors the local TIMs connected over MPI or Ethernet, maps their most important status displays on ST7sc tags, forwards received messages for message decoding or transfers the messages to be sent to the relevant TIM for WAN communication.

TCP/IP

(Transmission Control Protocol / Internet Protocol)

Network protocols for connecting two computers in the Internet. IP is the basic protocol. UDP is based on IP and sends individual packets. These can arrive at the recipient in a different order from the order in which they were sent - they can even be lost. TCP is used to make the connection reliable and makes sure, for example, that data packets are passed on to the application in the correct order. In addition to the IP addresses, UDP and TCP also involve port numbers between 1 and 65535 with which the various services can be distinguished.

On a Windows PC, the WINSOCK.DLL (or WSOCK32.DLL) is responsible for handling the two protocols.

TD7, TD7onCPU, TD7onTIM

→ *SINAUT TD7 Library*

TIM

(Telecontrol Interface Module)

The TIM transmission processor is a communications module that handles all data transmission functions provided by the SINAUT system independently. Depending on the type, the TIM has one or two WAN interfaces, an MPI interface or an Ethernet interface. Depending on the requirements, a variety of transmission equipment can be connected. The module is supplied in an S7-300 housing.

The TIM is available in two basic variants:

- TIM 4

  TIM 4R / TIM 4RD with MPI port both for installation as a CP in an S7-300 and as a stand-alone device for connection over MPI to one or more S7-400 and S7-300 PLCs. The TIM 4R / TIM 4RD has two WAN ports via serial interfaces for external modems. The two WANs can be identical or different, for example dedicated line plus telephone network.

  The TIM 4 series also includes the older modules TIM 42/42D, TIM 43/43D, TIM 44/44D and TIM 4V/4VD, see release 05/2007 of the manual (http://support.automation.siemens.com/WW/view/en/22548697).

- Ethernet TIM
  - TIM 3V-IE variants with two WAN attachments: An RS-232 port and an RJ-45 Ethernet port, only for installation as a CP in S7-300. With the TIM 3V-IE, SINAUT communication is either over the RS-232 or over the Ethernet port. The TIM 3V-IE Advanced can use the RS-232 port at the same time as the Ethernet port.
  - TIM 4R-IE with four WAN connections: Two RS-232 interfaces and two RJ-45 Ethernet interfaces. Installation as a CP in an S7-300 or as standalone device in a separate TIM rack. With the TIM 4R-IE, SINAUT communication is handled over the RS-232 and over the Ethernet interfaces simultaneously.

The TIM 3 series includes the older modules TIM 3V, TIM 32, TIM 33 and TIM 34, see release 05/2007 of the manual (http://support.automation.siemens.com/WW/view/en/22548697).

Topology

The topology describes the network structure. It specifies how a network (transmission medium and connectable devices or computers) is interconnected. Possible structures are linear (bus), star, ring, redundant ring and tree structure.

Transmission Control Protocol

Protocol for connection-oriented data transmission in networks; it belongs to the family of Internet protocols.

Unconditional spontaneous message

In the dial-up network, you can specify whether or not a change causes an "conditional spontaneous" or "unconditional spontaneous" transmission for each individual message.

Unconditional spontaneous messages cause the connection to be established immediately. Even with pay-by-volume/time transmission in a GPRS network, you can use the message prioritization "unconditional spontaneous"; in other words, in contrast to a "conditional spontaneous" message, a message is transmitted immediately.

See also → "conditional spontaneous message" and → "spontaneous message"

VPN

(Virtual Private Network)

Technology for secure transportation of confidential data in public IP networks, for example Internet.

WAN

(Wide Area Network)

Data network with a large geographical span, such as the Internet. telephone or enterprise networks.

The following WANs can be used with SINAUT:

- Classic WAN networks

  SINAUT communication is possible via the following networks:

  – Private or leased dedicated lines

  – Analog telephone network

  – Digital ISDN network

  – Private wireless networks (directional or omnidirectional wireless and private mobile wireless with time slots)

  – Mobile wireless network

  A classic WAN is connected via suitable transmission device (modem) to a serial interface of the TIM.

- IP-based WAN networks

  SINAUT communication is possible via the following IP-based networks:

  – Over wireless by using special wireless systems optimized for Ethernet, for example, the components of a SCALANCE W IWLAN system

  – Via fiber-optic cable, for example in conjunction with SCALANCE X switches with optical ports. This allows distances of up to 70 km to be covered.

  – Over public networks and Internet with DSL or GPRS

  – Private wideband systems such as OTN, PCM30 etc.

  An IP-based WAN is normally connected to an RJ-45 interface of an Ethernet TIM via an Ethernet-compliant module (see section "Installation guidelines").

  To allow use of GPRS, the serial interface of a station TIM 3V-IE can be connected to a GSM network via the GSM modem MD720-3. This requires the MSC protocol to be enabled in the STEP 7 > Properties dialog of the TIM > "Interfaces" tab. The WAN interface then behaves like a virtual Ethernet interface.

WinCC

WinCC is a cross-branch and technologically neutral system for the solution of visualization and process control tasks in production and process engineering. It provides function modules for graphics display (Graphics Designer), for signaling (Alarm Logging), for archiving (Tag Logging) and logging.

WinCC API

As a completely open and expandable system, WinCC makes a comprehensive API (Application Program Interface) available. This involves an interface over which the user programs such as ST7cc Server and ST7cc Config can access WinCC. A comprehensive description is available with the WinCC ODK (Open Developers Kit).

WinCC buffer

Message and archive processing can be assigned to the ST7cc tags. If this is the case, individual messages or archive data are generated in ST7cc, that are transferred over the ODK interface to Alarm Logging or Tag Logging for further processing.

The results of processing an ST7cc tag can, however, accrue faster than they can be accepted by WinCC. The WinCC buffer takes the WinCC jobs from the ST7cc processing and therefore separates the asynchronous procedures of job creation and job processing.

WinCC channel DLL

To allow WinCC to communicate with the widest variety of data sources (programmable controllers, ST7cc servers etc.), various communications drivers are used.

A communications driver is a C++ DLL that communicates with the data manager over an interface known as the channel API that is specified by the data manager. The WinCC tags are supplied with process values over the communications driver.

WinCC tag

WinCC tags are central elements to allow process values to be accessed. Within a WinCC project, they have a unique name and a data type. A WinCC tag is assigned to a logical connection that specifies which channel supplies the process values of the tags and over which connection.

For the WinCC tags whose data sources are the ST7 tags, the channel DLL is the connection over which the ST7cc server supplies the process values.

The WinCC tags required for ST7cc can be generated with ST7cc Config.

WinCC Tag Logging (Runtime)

Tag Logging is used to receive data from running processes and to prepare it for display and archiving. The data formats and the acquisition times and archiving times can be set as required.

WinCC Tag Logging is 'computer time'-oriented and [ ] not intended for the arrival of data with a delay offset as is the case with SINAUT ST7. This means that the ST7cc server must make certain archiving preparations for WinCC. The ST7cc server transfers the values to be archived to Tag Logging over the ODK interface. This ensures the chronological arrangement of the archive values even if process data is delivered by the ST7 stations, for example, with an offset of an hour.

WinCC tag management

WinCC tag management covers all WinCC tags.

WinCC Tag Management

Each element (in other words, every tag) used in WinCC is collected in Tag Management and managed there.

Index

C

Configuration
- Changing the, 16, 169
- Connections, 81
- Creating a new project, 17
- Hardware configuration, 21
- Network configuration, 17
- Networks, network attachments, 46
- Parameters for subscribers, 92
- Redundant ST7cc/sc control center, 89
- SMS Configuration, 104
- Subscriber data, 87
- TD7onTIM, 117
- Telephone number, 100

Configuration software, 13
Connection configuration, 82
Consistency check, 78, 164, 168
Current information, 4
Cyclic program OB1, 206

D

Data objects, 119
Data point typicals, 220
Diagnostic messages
- of TD7onCPU, 424
- of the TIM, 411

Diagnostics and Service tool
- Overview of functions, 333
- PG Routing, 332

E

Error program OB121, 213
Ethernet interface
- Static, 38
- Virtual, 38

F

FAQs, 4
Firmware update, 393

G

GPRS, 15, 38

H

HW Config, 21

I

Internet communication, 14, 15
- Simple, 38

Invalid Connections, 85

L

Lost connections, 86

M

Manuals, 4
Message protocol diagnostics
- ST7cc/ST7sc, 400
- Structure and function, 401
- Testcopy DB, 398
- TIM, 400

MSC communication
- Diagnostics, 355

MSC master station, 15
- Configuration, 112

MSC node station, 38
MSC protocol
- Enabling, 40

MSC station, 15, 38
- Configuration, 112

MSC tunnel, 15

N

NetPro, 17

O

Object communication, 199

Optional blocks, 290

P

PG Routing
Application, 442
Basic configurations, 430
Canceling the PG/PC assignment, 446
Introduction, 429
PG/PC assignment, 443
Range of functions, 433
System preparation, 435
Printing module information, 45

R

Receive channel, 119
Repair, 395

S

Send channel, 119
SIMATIC Manager, 17
SINAUT diagnostics
Block Structure for all CPUs, 374
SDB Viewer, 384
TD7 Block Structure, 369
TD7 Check of the Communication Configuration, 379
TD7 CPU Diagnostics, 368
TD7 CPU Program Comparison, 377
TD7onTIM diagnostics, 381
TIM diagnostic messages, 363
TIM Diagnostics, 345
TIM Message Monitor, 366
TIM subscriber diagnostics, 357
SINAUT objects, 118, 129, 199
SINAUT ST7 Configuration Tool, 80
SINAUT TD7, 177
Startup program OB100, 205
STEP 7 diagnostics
CPU messages, 335
General, 335
Module information, 337
Operating mode, 343
Setting the time, 344
Subscriber administration, 87
System data blocks
Generation of, 163
Transfer, 165
System function blocks SFBs, 327
System functions SFCs, 327
System objects, 119

T

TD7 basic blocks, 214
TD7 library
Block overview, 184
General, 181
Online help, 196
TD7onCPU, 177
TD7onTIM, 117
Test blocks, 321
Time synchronization
in SINAUT networks, 57
on the backplane bus of the S7-300, 29
on the Ethernet interface of the TIM, 29
on the MPI interface of the TIM, 29
with TD7onTIM, 160
Time-driven OB cyclic interrupt program, 212